

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

THIS PAGE BLANK (USPTO)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2000-172457
(P 2 0 0 0 - 1 7 2 4 5 7 A)
(43) 公開日 平成12年6月23日(2000.6.23)

(51) Int. Cl. ⁷	識別記号	F I	テーマコード (参考)
G06F 3/12		G06F 3/12	C 2C061
			A 5B021
B41J 29/38		B41J 29/38	Z

審査請求 未請求 請求項の数44 O L (全33頁)

(21) 出願番号 特願平10-342970

(22) 出願日 平成10年12月2日(1998.12.2)

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 志村 明弘

東京都大田区下丸子3丁目30番2号キヤノン株式会社内

(74) 代理人 100090538

弁理士 西山 恵三 (外2名)

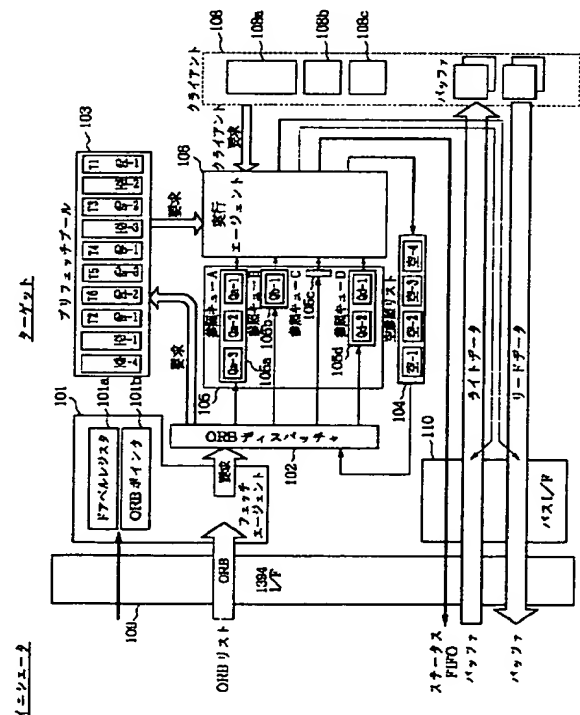
Fターム(参考) 2C061 AP03 AP07 HJ08 HQ20 HR02
5B021 AA01 AA02 BB00 BB01 CC04
CC05

(54) 【発明の名称】 通信制御方法、通信システム、印刷制御装置、印刷装置、ホスト装置、周辺装置、および、記憶媒体

(57) 【要約】

【課題】 送信する命令の数の管理を効率的に行うこと。

【解決手段】 イニシエータからターゲットに対して命令を発行することで、前記イニシエータの有する記憶領域に対して、前記ターゲットによりデータの書き込みあるいは読み出しを行わせてデータを交換する通信システムであって、前記イニシエータは、前記記憶領域に対する読み出し・書き込み命令を、ターゲットが保持できる命令の総和を超えないようにターゲットに送信し、前記ターゲットは、受信した読み出し・書き込み命令を保持し、それぞれの命令への参照を互いに異なるキューにより保持して、互いに独立に処理することを特徴とする。



【特許請求の範囲】

【請求項 1】 イニシエータからターゲットに対して命令を発行することで、前記イニシエータの有する記憶領域に対して、前記ターゲットによりデータの書き込みあるいは読み出しを行わせてデータを交換する通信制御方法であって、前記イニシエータは、前記記憶領域に対する読み出し・書き込み命令を、ターゲットが保持できる命令の総和を超えないようにターゲットに送信し、前記ターゲットは、受信した読み出し・書き込み命令を保持し、それぞれの命令への参照を互いに異なるキューにより保持して、互いに独立に処理することを特徴とする通信制御方法。

【請求項 2】 前記イニシエータからターゲットに対する命令発行は、前記イニシエータの有する記憶領域に命令を置き、前記イニシエータの指示によって前記ターゲットが前記イニシエータの記憶領域から読み出すことにより行われることを特徴とする請求項 1 に記載の通信制御方法。

【請求項 3】 前記キュー数の増加に応じて、前記ターゲットが保持できる命令の総和を増加させることを特徴とする請求項 1 または 2 に記載の通信制御方法。

【請求項 4】 前記キューの増加数以上の個数分、前記ターゲットが保持できる命令の総和を増加させることを特徴とする請求項 3 に記載の通信制御方法。

【請求項 5】 前記イニシエータの命令パラメータに従って前記ターゲットが保持できる命令の総和を増加させることを特徴とする請求項 1、2 または 3 に記載の通信制御方法。

【請求項 6】 前記キュー数の減少に応じて、前記ターゲットが保持できる命令の総和を減少させることを特徴とする請求項 1 または 2 に記載の通信制御方法。

【請求項 7】 キュー数の増加時に伴って増加した前記ターゲットが保持できる命令の総和数の増加分だけ、当該キューの減少時に前記ターゲットが保持できる命令の総和数を減少させることを特徴とする請求項 6 に記載の通信制御方法。

【請求項 8】 前記イニシエータの命令パラメータに従って前記ターゲットが保持できる命令の総和を減少させることを特徴とする請求項 6 に記載の通信制御方法。

【請求項 9】 前記ターゲット及びイニシエータは、前記書き込み・読み出し命令についてキュー識別子を付与し、キュー識別子毎に各命令を独立して処理することを特徴とする請求項 1 乃至 8 に記載の通信制御方法。

【請求項 1 0】 前記イニシエータは前記各キューに対して少なくとも 1 つの命令を前記ターゲットが保持できるように、各キューに対する命令数の割り当てを行うことを特徴とする請求項 1 乃至 9 に記載の通信制御方法。

【請求項 1 1】 前記イニシエータはキューの増加減少時各キューに対する命令数の割り当てを行うことを特徴とする請求項 1 0 に記載の通信制御方法。

【請求項 1 2】 イニシエータからターゲットに対して命令を発行することで、前記イニシエータの有する記憶領域に対して、前記ターゲットによりデータの書き込みあるいは読み出しを行わせてデータを交換する通信システムであって、前記イニシエータは、前記記憶領域に対する読み出し・書き込み命令を、ターゲットが保持できる命令の総和を超えないようにターゲットに送信し、前記ターゲットは、受信した読み出し・書き込み命令を保持し、それぞれの命令への参照を互いに異なるキューにより保持して、互いに独立に処理することを特徴とする通信システム。

【請求項 1 3】 前記イニシエータからターゲットに対する命令発行は、前記イニシエータの有する記憶領域に命令を置き、前記イニシエータの指示によって前記ターゲットが前記イニシエータの記憶領域から読み出すことにより行われることを特徴とする請求項 1 2 に記載の通信システム。

【請求項 1 4】 前記キュー数の増加に応じて、前記ターゲットが保持できる命令の総和を増加させることを特徴とする請求項 1 2 または 1 3 に記載の通信システム。

【請求項 1 5】 前記キューの増加数以上の個数分、前記ターゲットが保持できる命令の総和を増加させることを特徴とする請求項 1 4 に記載の通信システム。

【請求項 1 6】 前記イニシエータの命令パラメータに従って前記ターゲットが保持できる命令の総和を増加させることを特徴とする請求項 1 1、1 3 または 1 4 に記載の通信システム。

【請求項 1 7】 前記キュー数の減少に応じて、前記ターゲットが保持できる命令の総和を減少させることを特徴とする請求項 1 2 または 1 3 に記載の通信システム。

【請求項 1 8】 キュー数の増加時に伴って増加した前記ターゲットが保持できる命令の総和数の増加分だけ、当該キューの減少時に前記ターゲットが保持できる命令の総和数を減少させることを特徴とする請求項 1 7 に記載の通信システム。

【請求項 1 9】 前記イニシエータの命令パラメータに従って前記ターゲットが保持できる命令の総和を減少させることを特徴とする請求項 1 7 に記載の通信システム。

【請求項 2 0】 前記ターゲット及びイニシエータは、前記書き込み・読み出し命令についてキュー識別子を付与し、キュー識別子毎に各命令を独立して処理することを特徴とする請求項 1 2 乃至 1 9 に記載のシステム。

【請求項 2 1】 前記イニシエータは前記各キューに対して少なくとも 1 つの命令を前記ターゲットが保持できるように、各キューに対する命令数の割り当てを行うことを特徴とする請求項 1 2 乃至 2 0 に記載の通信システム。

【請求項 2 2】 前記イニシエータはキューの増加減少時各キューに対する命令数の割り当てを行うこ

とを特徴とする請求項 2 1 に記載の通信システム。

【請求項 2 3】 請求項 1 乃至 1 1 のいずれかに記載の通信制御方法を用いて、ホスト装置をイニシエータとし、プリンタ装置をターゲットとしてホスト装置とプリンタ装置とを接続し、前記プリンタ装置は、ホスト装置から印刷データを受信してそれを基に印刷出力し、前記ホスト装置は、プリンタ装置からのプリンタ状態情報を受信することを特徴とする印刷システム。

【請求項 2 4】 メモリ内の記憶領域に対して、通信により接続されたターゲットからデータを読み書きすることでターゲットとデータを交換する通信制御方法であって、アプリケーションからの要求に応じて命令をキューに登録するキューイング工程と、前記ターゲットから与えられた前記ターゲットが保持できる命令の総和のパラメータを超えないようにターゲットに対して命令を生成して送信する命令生成工程と、を備えることを特徴とする通信制御方法。

【請求項 2 5】 前記命令生成工程において前記各キューに対して少なくとも 1 つの命令を前記ターゲットが保持できるように、各キューに対する命令数の割り当てを行うことを特徴とする請求項 2 4 に記載の通信制御方法。

【請求項 2 6】 請求項 2 4 または 2 5 に記載の通信制御方法によりターゲットに印刷データを送信し、ターゲットから状態情報を受信することを特徴とする印刷制御装置。

【請求項 2 7】 通信により接続されたイニシエータの有する記憶領域に対してデータを読み書きしてイニシエータとデータを交換する通信制御方法であって、前記イニシエータからの命令を所定容量の記憶領域に登録する格納工程と、前記格納工程によって格納された命令を参照する参照データをキューに登録するキューイング工程と、前記キューで参照される命令を順次取り出して実行する順次実行工程と、を備えることを特徴とする通信制御方法。

【請求項 2 8】 前記キューの増加・減少に応じて前記所定容量の記憶領域の容量を増加・減少する容量変更工程と、増加分をイニシエータに通知する通知工程と、を備えることを特徴とする請求項 2 7 に記載の通信制御方法。

【請求項 2 9】 前記容量変更工程において、前記キューの増加数以上の個数分、前記ターゲットが保持できる命令の総和を増加させることを特徴とする請求項 2 8 に記載の通信制御方法。

【請求項 3 0】 前記容量変更工程において、前記イニシエータからの命令パラメータに応じて、前記ターゲットが保持できる命令の総和を増加させることを特徴とする請求項 2 8 に記載の通信制御方法。

【請求項 3 1】 前記容量変更工程において、キュー数の増加時に伴って増加した前記ターゲットが保持できる

命令の総和数の増加分だけ、当該キューの減少時に前記ターゲットが保持できる命令の総和数を減少させることを特徴とする請求項 2 8 乃至 3 0 に記載の通信制御方法。

【請求項 3 2】 前記容量変更工程において、前記イニシエータからの命令パラメータに応じて、前記ターゲットが保持できる命令の総和を減少させることを特徴とする請求項 2 8 に記載の通信制御方法。

【請求項 3 3】 請求項 2 7 乃至 3 2 に記載の通信制御方法によりイニシエータから印刷データを受信し、イニシエータに状態情報を送信することを特徴とする印刷装置。

【請求項 3 4】 要求を格納する格納手段と前記格納手段に格納された要求を参照する情報を複数の参照キューにキューイングする手段と有する周辺装置に接続されるホスト装置であって、クライアントからの要求に対応した複数の要求キューを保持する保持手段と、前記周辺装置の格納手段に格納可能な数を管理する管理手段と、前記管理手段に管理されている数に基づき、前記保持手段に保持された各キューの要求をリストに格納する手段とを有することを特徴とするホスト装置。

【請求項 3 5】 ホスト装置に接続される周辺装置であって、前記ホスト装置からの要求を格納する格納手段と、前記格納手段に格納された要求を参照する情報を複数の参照キューにキューイングする手段と、前記参照する情報に基づきそれぞれの参照キューの要求を独立して実行する手段とを有することを特徴とする周辺装置。

【請求項 3 6】 前記周辺装置は、プリンタであり、前記複数の参照キューには、印刷ジョブ、フィニッシャー制御コマンドがキューイングされることを特徴とする請求項 3 6 に記載の周辺装置。

【請求項 3 7】 要求を格納する格納手段と前記格納手段に格納された要求を参照する情報を複数の参照キューにキューイングする手段と有する周辺装置に接続されるホスト装置の制御方法であって、クライアントからの要求に対応した複数の要求キューを保持手段に保持するステップと、前記周辺装置の格納手段に格納可能な数を超えないように前記保持手段に保持された各キューの要求をリストに格納するステップとを有することを特徴とする制御方法。

【請求項 3 8】 ホスト装置に接続される周辺装置の制御方法であって、前記ホスト装置からの要求を格納手段にするステップと、前記格納手段に格納された要求を参照する情報を複数の参照キューにキューイングするステップと、前記参照する情報に基づきそれぞれの参照キューの要求を独立して実行するステップとを有することを特徴とする制御方法。

【請求項 3 9】 前記周辺装置は、プリンタであり、前記複数の参照キューには、印刷ジョブ、フィニッシャー制御コマンドがキューイングされることを特徴とする請

求項 3 8 記載の周辺装置。

【請求項 4 0】 要求を格納する格納手段と前記格納手段に格納された要求を参照する情報を複数の参照キューにキューイングする手段と有する周辺装置に接続されるホスト装置の制御プログラムを記憶する記憶媒体であって、クライアントからの要求に対応した複数の要求キューを保持手段に保持するステップと、前記周辺装置の格納手段に格納可能な数を超えないように前記保持手段に保持された各キューの要求をリストに格納するステップとを有するプログラムを記憶することを特徴とする記憶媒体。

【請求項 4 1】 ホスト装置に接続される周辺装置の制御プログラムを記憶する記憶媒体であって、前記ホスト装置からの要求を格納手段にするステップと、前記格納手段に格納された要求を参照する情報を複数の参照キューにキューイングするステップと、前記参照する情報に基づきそれぞれの参照キューの要求を独立して実行するステップとを有するプログラムを記憶することを特徴とする記憶媒体。

【請求項 4 2】 前記周辺装置は、プリンタであり、前記複数の参照キューには、印刷ジョブ、フィニッシャー制御コマンドがキューイングされることを特徴とする請求項 4 1 記載の記憶媒体。

【請求項 4 3】 ホスト装置と周辺装置とからなるシステムであって、前記ホスト装置は、クライアントからの要求に対応した複数の要求キューを保持する保持手段と、前記周辺装置の格納手段に格納可能な数を管理する管理手段と、前記管理手段に管理されている数に基づき、前記保持手段に保持された各キューの要求をリストに格納する手段とを有し、前記周辺装置は、前記ホスト装置からの要求を格納する格納手段と、前記格納手段に格納された要求を参照する情報を複数の参照キューにキューイングする手段と、前記参照する情報に基づきそれぞれの参照キューの要求を独立して実行する手段とを有することを特徴とするシステム。

【請求項 4 4】 前記周辺装置は、プリンタであり、前記複数の参照キューには、印刷ジョブ、フィニッシャー制御コマンドがキューイングされることを特徴とする請求項 4 3 記載のシステム。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】 本発明は、例えば、イニシエータ（ホストコンピュータ）とターゲット（プリンタ等）の機器同士を接続する通信制御方法及びシステムに関する。

【 0 0 0 2 】

【従来の技術】 近年、IEEE 1394 インターフェースが、コンピュータと周辺機器、あるいは周辺機器同士の接続に用いられつつある。IEEE 1394 インターフェースは、セントロニクスインターフェースなどのハ

ンドシェイク方式に比べて高速で、しかも双方向通信が可能である。また、メモリバスモデルのインターフェースであり、IEEE 1394 インターフェースで接続された機器は、相手の機器に対して指定されたアドレスのデータを読み書きすることができる。

【 0 0 0 3 】 この IEEE 1394 は、広範囲に応用するための物理層及びリンク層のプロトコルを定めたもので、機器ごとの詳細なプロトコルは定められていない。そのため、物理・リンク層として IEEE 1394 を利用した SBP (Serial Bus Protocol) - 2 などのトランスポート層のプロトコルが提案されている。トランスポート層はアプリケーションに対してデータ転送機能を提供する層であり、この層を利用するアプリケーションは、互いにデータの交換が可能となる。

【 0 0 0 4 】 この SBP - 2 なるプロトコルは、IEEE 1394 のメモリバスモデルとしての特徴を活かしたプロトコルであり、データの受信側がそれ自身の都合に応じてデータを受信することができる。これに対して、SBP - 2 以外のプロトコルには、非同期に発生するデータの転送が可能となるものも、マルチチャネルが実現できるものもあるが、そのようなプロトコルでは IEEE 1394 のメモリバスモデルとしての特徴を活かすことができない。すなわち、ホストとプリンタの通信であれば、プリンタ側の都合に応じてデータ転送を行うことができず、ホストがプリンタの状態を監視しながらデータ転送を行わねばならないことになる。

【 0 0 0 5 】 SBP - 2 では、データを転送する場合に、まず送信側でログインという操作を行って通信相手との間のチャネルを確立する。この際、ログインした側がイニシエータと呼ばれ、イニシエータと接続された相手側がターゲットと呼ばれる。データ転送は、イニシエータからの指示に応じて、ターゲットからイニシエータのバッファにデータを読み書きすることで行われる。この方式においては、イニシエータは、送信するデータを格納したバッファのアドレスやサイズ等が書かれた ORB (Operation Request Block) を作成し、その ORB のアドレスをターゲットに知らせる。ターゲットでは自分の都合に合わせて ORB に書かれたアドレスやサイズに基づいてイニシエータからデータを読み出し、あるいはデータを書き込み、それらの処理後ステータスブロックを作成して、処理の状態をイニシエータに知らせることになる。

【 0 0 0 6 】

【発明が解決しようとする課題】 この IEEE 1394 の上に構築される SBP - 2 プロトコルを利用して通信を行う場合、特に、ホストコンピュータ等のデータ源をイニシエータとし、それからターゲットであるプリンタ装置などの周辺機器へのデータ転送に応用する場合、次のような 4 つの問題点があった。

【 0 0 0 7 】 （問題点 1）全 2 重通信のために手順が複

雑である。

【0008】SBP-2では、基本的にデータの転送はイニシエータによって管理され、ターゲットはイニシエータへの非同期のデータ転送を行うことができない。すなわち、SBP-2では、ターゲットがイニシエータへのデータ転送を行いたい時には、イニシエータに対してアンソリシテッドステータスでデータリード要求を送り、イニシエータはそれに応じてORB生成し、ペンディング中のORB（イニシエータからターゲットへのデータ転送要求などが含まれている）のリストの末尾にその生成したORBをつける。このORBは先頭から順番に処理されるため、ターゲットがイニシエータにリード要求を発行したタイミングではなく、イニシエータ側のORBの処理が進み、ターゲットからのデータリード要求によるORBが処理されたときに初めてターゲットからイニシエータにデータが転送されることになる。すなわち、双方向の非同期なデータ転送が行えず、ターゲットからイニシエータに転送すべきデータが非同期に発生する場合、例えばターゲットがプリンタであれば、そのプリンタでエラーが発生する場合など、直ちにイニシエータに伝達すべきデータを即時に伝達できない。

【0009】このため、例えばプリンタで非同期に発生するデータを即座にホストに転送するためには、プリンタをイニシエータとしてログイン手続きを行い、ホストコンピュータをターゲットとするデータ転送を行わねばならない。

【0010】このようにホストコンピュータとプリンタで互いにログインしてそれぞれがイニシエータでありターゲットであるような状況では、ホストコンピュータ及びプリンタの双方にイニシエータとしてのプロセスとターゲットとしてのプロセスを備えなければならない。また、ログインの操作もプリンタから行わねばならない。プリンタのような画像を扱う周辺装置では、画像処理のために大量のメモリ資源やプロセッサ資源を消費する。そのために、装置の構成を簡略化して原価を節減したり、処理を迅速に行うため画像処理用途以外に用いられる資源をできるだけ節約しなければならない。しかし、上述のように多くのプロセスを稼働することになればその分多くの資源を消費することになり、原価低減・処理の効率化という目的に反することになる。

【0011】また、ホストコンピュータとプリンタとの関係であれば、各方向に流れるデータは、印刷データとそれに対する処理状況といったように、互いに関連付けられるものであるが、各方向について独立したログインによりチャンネルを設定すると、それらデータとレスポンスとを関連づけなければならない、そのための処理手順を新たに追加する必要がある。

【0012】このようにIEEE1394及びSBP-2をそのままホストコンピュータ・プリンタ装置間の通信に適用することは適切でなく、各装置において必要と

される資源を減らすことや効率を向上させることが難しかった。

【0013】（問題点2）マルチチャンネルを実現できない。

【0014】最近、周辺装置として種々の機能を複合させた複合機が利用されつつある。例えば、ファクシミリ装置を、スキャナ単体、プリンタ単体、ファクシミリとしホストコンピュータ等から利用できるデジタル複合機などがある。このような装置を利用する際には、各単体機能ごとに独立した複数のチャンネルを介して通信を行えば、同時に複数の機能を利用することができる。

【0015】しかしながら、SBP-2では、マルチチャンネルを提供できないため、そのように単体機能を同時に利用することが難しい。

【0016】（問題点3）バスリセットに対応できない。

【0017】IEEE1394では、1394シリアルバスへの機器の新たな接続や取り外し、あるいは、接続された機器の電源投入や切断といった、ネットワーク構成の変化要因となる状態変化が発生するとバスリセットが発生する。バスリセットは、バス上で上記のような状態変化を検知したノードがバスリセット信号をバス上に送信することで発生する。発生されたバスリセット信号は、ノードからノードへと伝達され、ネットワーク上の全ノードがバスリセット信号を受信すると、バスリセットのための一連の動作が各ノードで行われる。

【0018】このようにバスリセットは、ネットワーク上のノードにおける処理とは非同期に発生する。また、バスリセットの原因となったノードとアプリケーションについては無関係なノードであっても、一旦バスリセットが発生してしまえばバスリセットの処理を行わねばならない。このバスリセットの過程で、SBP-2で通信を行っているノードでは、設定されていたコネクションが切断されてしまい、再度コネクションをつなげてもバスリセット直前の状態から処理を続行できる保証が与えられていない。

【0019】また、イニシエータがターゲットに各キュー毎に数を超えないよう命令を送信していたため、使われないキューに確保されている数が無駄になる事がある。

【0020】本発明は上記従来例に鑑みてなされたもので、1回のログインで全2重通信（互いに非同期な双方向通信）を可能とし、また、データの交換に必要なプロセスやメモリといった資源を効率的に利用できる通信制御方法及び装置及びそれを用いた印刷装置を提供することを目的とする。

【0021】また、マルチチャンネルを実現する通信制御方法及び装置及びそれを用いた印刷装置を提供することを目的とする。

【0022】また、バスリセットが発生しても、バスリ

セット直前の状態から処理の続行を保証する通信制御方法及び装置及びそれを用いた印刷装置を提供することを目的とする。

【0023】また、イニシエータからターゲットに送信できる命令の数の管理をターゲットの各キュー毎に行うのではなく、一元管理することにより効率的にリソースを利用できる通信制御方法、通信システム、印刷制御装置、印刷装置、ホスト装置、周辺装置、および、記憶媒体を提供することを目的とする。

【0024】また、キューによる多重化路におけるターゲットのコマンドプール領域全体の割り当てをイニシエータが動的に行うことによって、通信効率及びターゲットのリソース利用効率が向上するとともに、コネクションに用いられるキュー数に応じてターゲットのコマンドプール領域の増減が行われることにより、ターゲットのリソース利用効率が向上する通信制御方法、通信システム、印刷制御装置、印刷装置、ホスト装置、周辺装置、および、記憶媒体を提供することを目的とする。

【0025】

【課題を解決するための手段】上記目的を達成するために本発明は次のような構成から成る。すなわち、イニシエータからターゲットに対して命令を発行することで、前記イニシエータの有する記憶領域に対して、前記ターゲットによりデータの書き込みあるいは読み出しを行わせてデータを交換する通信制御方法であって、前記イニシエータは、前記記憶領域に対する読み出し・書き込み命令を、ターゲットが保持できる命令の総和を超えないようにターゲットに送信し、前記ターゲットは、受信した読み出し・書き込み命令を保持し、それぞれの命令への参照を互いに異なるキューにより保持して、互いに独立に処理することの特徴とする。

【0026】また、本発明の通信システムは、イニシエータからターゲットに対して命令を発行することで、前記イニシエータの有する記憶領域に対して、前記ターゲットによりデータの書き込みあるいは読み出しを行わせてデータを交換する通信システムであって、前記イニシエータは、前記記憶領域に対する読み出し・書き込み命令を、ターゲットが保持できる命令の総和を超えないようにターゲットに送信し、前記ターゲットは、受信した読み出し・書き込み命令を保持し、それぞれの命令への参照を互いに異なるキューにより保持して、互いに独立に処理することの特徴とする。

【0027】また、本発明の印刷システムは、前記通信制御方法を用いて、ホスト装置をイニシエータとし、プリンタ装置をターゲットとしてホスト装置とプリンタ装置とを接続し、前記プリンタ装置は、ホスト装置から印刷データを受信してそれを基に印刷出力し、前記ホスト装置は、プリンタ装置からのプリンタ状態情報を受信することの特徴とする。

【0028】また、本発明の通信制御方法は、メモリ内

の記憶領域に対して、通信により接続されたターゲットからデータを読み書きすることでターゲットとデータを交換する通信制御方法であって、アプリケーションからの要求に応じて命令をキューに登録するキューイング工程と、前記ターゲットから与えられた前記ターゲットが保持できる命令の総和のパラメータを超えないようにターゲットに対して命令を生成して送信する命令生成工程とを備えることを特徴とする。

【0029】また、本発明の印刷制御装置は、前記通信制御方法によりターゲットに印刷データを送信し、ターゲットから状態情報を受信することの特徴とする。

【0030】また、本発明の通信制御方法は、通信により接続されたイニシエータの有する記憶領域に対してデータを読み書きしてイニシエータとデータを交換する通信制御方法であって、前記イニシエータからの命令を所定容量の記憶領域に登録する格納工程と、前記格納工程によって格納された命令を参照する参照データをキューに登録するキューイング工程と、前記キューで参照される命令を順次取り出して実行する順次実行工程と、を備えることを特徴とする。

【0031】また、本発明の印刷装置は、前記通信制御方法によりイニシエータから印刷データを受信し、イニシエータに状態情報を送信することの特徴とする。

【0032】また、本発明のホスト装置は、要求を格納する格納手段と前記格納手段に格納された要求を参照する情報を複数の参照キューにキューイングする手段と有する周辺装置に接続されるホスト装置であって、クライアントからの要求に対応した複数の要求キューを保持する保持手段と、前記周辺装置の格納手段に格納可能な数を管理する管理手段と、前記管理手段に管理されている数に基づき、前記保持手段に保持された各キューの要求をリストに格納する手段とを有することの特徴とする。

【0033】また、本発明の周辺装置は、ホスト装置に接続される周辺装置であって、前記ホスト装置からの要求を格納する格納手段と、前記格納手段に格納された要求を参照する情報を複数の参照キューにキューイングする手段と、前記参照する情報に基づき要求を実行する手段とを有することの特徴とする。

【0034】また、本発明の制御方法は、要求を格納する格納手段と前記格納手段に格納された要求を参照する情報を複数の参照キューにキューイングする手段と有する周辺装置に接続されるホスト装置の制御方法であって、クライアントからの要求に対応した複数の要求キューを保持手段に保持するステップと、前記周辺装置の格納手段に格納可能な数を超えないように前記保持手段に保持された各キューの要求をリストに格納するステップとを有することの特徴とする。

【0035】また、本発明の制御方法は、ホスト装置に接続される周辺装置の制御方法であって、前記ホスト装置からの要求を格納手段にするステップと、前記格納手

段に格納された要求を参照する情報を複数の参照キューにキューイングするステップと、前記参照する情報に基づき要求を実行するステップとを有することを特徴とする。

【0036】また、本発明の記憶媒体は、要求を格納する格納手段と前記格納手段に格納された要求を参照する情報を複数の参照キューにキューイングする手段と有する周辺装置に接続されるホスト装置の制御プログラムを記憶する記憶媒体であって、クライアントからの要求に対応した複数の要求キューを保持手段に保持するステップと、前記周辺装置の格納手段に格納可能な数を超えないように前記保持手段に保持された各キューの要求をリストに格納するステップとを有するプログラムを記憶することを特徴とする。

【0037】また、本発明の記憶媒体は、ホスト装置に接続される周辺装置の制御プログラムを記憶する記憶媒体であって、前記ホスト装置からの要求を格納手段にするステップと、前記格納手段に格納された要求を参照する情報を複数の参照キューにキューイングするステップと、前記参照する情報に基づき要求を実行するステップとを有するプログラムを記憶することを特徴とする。

【0038】

【発明の実施の形態】（第1の実施の形態）本発明の第1の実施の形態として、ホストコンピュータとプリンタとを、IEEE1394で接続し、その上に構築されたSBP-2プロトコルを用いた、本発明に係るプロトコル（以下SHPT-2と呼ぶ）に従ってデータ転送を行うプリンタシステムを説明する。図24は、そのプリンタシステムにおけるハードウェア構成図である。

【0039】＜システムのハードウェア構成＞図24において、ホストコンピュータ200は、ROM3のプログラム用ROMに記憶された文書処理プログラム等に基づいて、図形、イメージ、文字、表（表計算等を含む）等が混在した文書処理を実行するCPU1を備える。CPU1はシステムデバイス4に接続される各デバイスをCPU1が総括的に制御する。また、ROM3のプログラム用ROMには、CPU1の制御プログラム等が記憶され、ROM3のフォント用ROMには上記文書処理の際に使用するフォントデータ等が記憶され、ROM3のデータ用ROMは上記文書処理等を行う際に使用する各種データが記憶される。RAM2は、CPU1の主メモリ、ワークエリア等として機能する。

【0040】プログラムはRAM2に格納されても良い。また、RAM2には、送信データバッファやORBを格納するシステムメモリが確保される。

【0041】キーボードコントローラ（KBC）5は、キーボード9や不図示のポインティングデバイスからのキー入力を制御する。CRTコントローラ（CRTC）6は、CRTディスプレイ（CRT）10の表示を制御する。メモリコントローラ（MC）7は、ブートプログ

ラム、種々のアプリケーション、フォントデータ、ユーザファイル、編集ファイル等を記憶するハードディスク（HD）、フロッピーディスク（FD）等の外部メモリ11とのアクセスを制御する。1394インターフェース8は、IEEE1394規格に従ってプリンタ100に接続されて、プリンタ100との通信制御処理を実行する。なお、CPU1は、例えばRAM2上に設定された表示情報RAMへのアウトラインフォントの展開（ラスタライズ）処理を実行し、CRT10上でのWYSIWYGを可能としている。また、CPU1は、CRT10上の不図示のマウスカーソル等で指示されたコマンドに基づいて登録された種々のウィンドウを開き、種々のデータ処理を実行する。

【0042】プリンタ100において、プリンタCPU12は、ROM13のプログラム用ROMに記憶された制御プログラム等或いは外部メモリ14に記憶された制御プログラム等に基づいてシステムバス15に接続される各種のデバイスとのアクセスを総括的に制御し、印刷部インタフェース16を介して接続される印刷部（プリンタエンジン）17に出力情報としての画像信号を出力する。また、このROM13のプログラムROMには、後述する各種エージェントや画像処理を実現するCPU12の制御プログラム等が記憶される。ROM13のフォント用ROMには上記出力情報を生成する際に使用するフォントデータ等が記憶され、ROM13のデータ用ROMには、ハードディスク等の外部メモリ14が無いプリンタの場合には、ホストコンピュータ上で利用される情報等が記憶されている。CPU12は1394インターフェース18を介してホストコンピュータとの通信処理が可能となっており、プリンタ内の情報等をホストコンピュータ200に通知可能に構成されている。

【0043】RAM19は、CPU12の主メモリ、ワークエリア等として機能するRAMで、図示しない増設ポートに接続されるオプションRAMによりメモリ容量を拡張することができるように構成されている。なお、RAM19は、出力情報展開領域、環境データ格納領域、NVRAM等に用いられる。

【0044】前述したハードディスク（HD）、ICカード等の外部メモリ14は、メモリコントローラ（MC）20によりアクセスを制御される。外部メモリ14は、オプションとして接続され、フォントデータ、エミュレーションプログラム、フォームデータ等を記憶する。また、操作パネル18には、操作のためのスイッチおよびLED表示器等が配されている。また、前述した外部メモリは1個に限らず、少なくとも1個以上備え、内蔵フォントに加えてオプションフォントカード、言語系の異なるプリンタ制御言語を解釈するプログラムを格納した外部メモリを複数接続できるように構成されていても良い。さらに、図示しないNVRAMを有し、操作パネル1012からのプリンタモード設定情報を記憶す

るようにしても良い。

【0045】また、付図示ではあるが、ステイブル機能、ソーター機能等を有するフィニッシャーを接続可能である。

【0046】＜イニシエータの構成＞上述したハードウェア構成において、プリンタ100をターゲットとし、ホストコンピュータ200をイニシエータとした通信システムを、図1及び図2に示す。本実施形態に置いては、これら構成は、それぞれホストコンピュータ及びプリンタにおけるCPUによりプログラムを実行することで実現される。まず、図2のイニシエータから説明する。

【0047】図2において、イニシエータであるホストコンピュータにおいては、プリンタドライバ等のクライアント206はSHPT-2処理部201を介してプリンタに対するデータ転送要求を発行するとともに、プリンタからの応答を受ける。

【0048】SHPT-2処理部201は、システムメモリに生成されるI/O要求キューやORBを管理している。クライアント206は単数或いは複数のクライアントプロセスからなり、各クライアントプロセスからのI/O要求コマンドは各クライアントプロセスの通信に用いられる各I/O要求キューにそれぞれキューイングされる。図中、4つのI/O要求キューが例示されているが、I/O要求キューの数はこれに限られるものではなく、I/O要求キューは各クライアントプロセスの通信接続時に動的に用意され、各クライアントプロセスの通信には単数或いは複数のI/O要求キューが用いられる。

【0049】図中の例では、クライアントプロセス206aの通信には2つのI/O要求キュー202a、202bが用いられ、I/O要求キュー202aにはクライアントプロセス206aからのライト要求コマンドが、I/O要求キュー202bにはクライアントプロセス206aからのリード要求コマンドが、それぞれキューイングされ、これらの二つのI/O要求キューにより非同期全二重通信が行なわれる様子を示している。同様にクライアントプロセス206bの通信にはI/O要求キュー202cが用いられ、このI/O要求キューにライト要求コマンドがキューイングされ、単方向通信が行われ、また、クライアントプロセス206cの通信にはI/O要求キュー202dが用いられ、このI/O要求キューにライト要求コマンド及びリード要求コマンドの一方もしくは両方をキューイングすることで同期半二重通信が行なわれる様子を示している。

【0050】ORBはイニシエータであるホストコンピュータからターゲットであるプリンタへ、あるいはプリンタからホストコンピュータへ渡されるデータバッファのアドレスやサイズ等を格納したブロックであり、先頭のORBからORBリスト204に順次リンクされてい

る。このORBについては、次のような処理の原則がある。

【0051】(1) 各I/O要求キューそれぞれの先頭から取り出したコマンドから順にORBが作成され、ORBリストの末尾につけられる。コマンドをORBリストの末尾につける順序には特に制約はない。各I/O要求キューに優先順位を付け、特定のI/O要求キューの先頭のコマンドが優先的にORBリストの末尾につけられるようにしてもよい。

【0052】(2) ORBは、ORBリストの先頭から順次フェッチされる。完了通知(ステータスブロック)を受信すると、そのステータスに対応するORBはORBリストからはずされる。

【0053】(3) リンクされるORBの数の上限は、後述するターゲットにおけるプリフェッチブールの容量(それぞれ処理中のコマンドも含む)と同一である。

【0054】(1)、(2)項により、各I/O要求キュー毎のI/O要求は、それぞれのI/O要求において、発生した順にターゲットに発行されることが保証される。また、(3)項により、ORBリスト中のI/O要求はターゲットに渡されていることが保証される。この(3)項を実現するために、SHPT-2処理部201は、各要求キューに対してそれぞれ1つのカウンタ及びプリフェッチブールに対して1つのカウンタを用意している。

【0055】各I/O要求キューに対するカウンタはCurrent-QUEというカウンタであり、図の例では4つのI/O要求キューに対して4つのCurrent-QUEカウンタ203a、203b、203c、203dが示されている。それぞれのCurrent-QUEカウンタは、分母(右側)にI/O要求キューに割り当てられたプリフェッチブール内における予約域の数、及び、分子(左側)にそのI/O要求キューに属するコマンドの現在のORBリスト内での個数を保持している。

【0056】図中の例は、各I/O要求キューに割り当てられたプリフェッチブール内における予約域の数が“1”であり、各要求キュー202a、202b、202c、202dに属するコマンドの現在のORBリスト内での個数がそれぞれ“3”、“1”、“0”、“2”である状態を示している。

【0057】プリフェッチブールに対するカウンタはCurrent-POOLというカウンタ203eであり、分母(右側)にはプリフェッチブールの全域の数“10”からそれぞれのI/O要求キューの予約域の数“1+1+1+1”の総和“4”を減じたもの“6”がセットされる。このカウンタは各I/O要求キューに割り当てられた予約域以外のプリフェッチブールにおける空数を示している。

【0058】また、分子(左側)には、要求キューに貸し出せる数“3”がセットされる。具体的には、203

a ~ 203d の分母より分子の大きいものから、この場合、203a と 203d の分子 - 分母 ($3 - 1 = 2$ 、 $2 - 1 = 1$) を計算し、各 I/O 要求キューに割り当てられた予約域以外のプリフェッチブルにおける空数 6 から計算された 2 と 1 を加えた 3 を減じたもの “3” がセットされる。

【0059】図中の例は各 I/O 要求キューに割り当てられた予約域以外のプリフェッチブルにおける空数が “6” である状態を示している。

【0060】このプリフェッチブルの全域の数はターゲットとなる機器が保持しており、ログイン時等にターゲットから読み出され、イニシエータに記憶される。

【0061】図中の例は、プリフェッチブルの全域の数が “10” である状態を示している。各 I/O 要求キューに付随する Current-QUE カウンタの現在の ORB リスト内での個数は ORB の生成と削除に合わせて増減される。

【0062】ORB の生成に当たっては、Current-QUE カウンタの現在の ORB リスト内での個数が I/O 要求キューに初期に割り当てられたプリフェッチブル内における予約域の数を超えない場合には、Current-QUE カウンタを増加させ、ORB を ORB リストに追加する。そうでない場合には Current-POOL カウンタが “1” 以上であるかどうかを調べ、“1” 以上であれば、Current-POOL カウンタを減じ、Current-QUE カウンタを増加させ、ORB を ORB リストに追加する。Current-POOL カウンタが既に “0” になっている場合には ORB を ORB リストに追加しない。

【0063】ORB の削除にあたっては、Current-QUE カウンタの現在の ORB リスト内での個数が I/O 要求キューに初期に割り当てられたプリフェッチブル内における予約域の数を超えている場合には、Current-POOL カウンタを増加させる。いずれの場合にも Current-QUE カウンタの現在の ORB リスト内での個数は減じられる。

【0064】また、ORB を生成した際には、ターゲット側のドアベルレジスタと呼ばれるレジスタに所定値を書き込む、あるいは、リストの先頭の ORB のアドレスを ORB ポインタと呼ばれるレジスタに書き込むことにより、ORB が発生したことをターゲットに知らせる。既にターゲットが実行している ORB リストの最後に ORB を追加した場合にはドアベルレジスタへの書き込みを行い、ORB リストがない状態で新規に先頭の ORB を作成した場合には、ORB ポインタへの書き込みを行う。この手続きは SBP-2 で規定されている。

【0065】また、SHPT-2 処理部 201 は、ステータス FIFO 205 を含む。

【0066】1394 インターフェース 109 を介して受信されたステータスは、SHPT-2 処理部 201 により処理される。SHPT-2 処理部 201 は、受信し

たステータスに対応する ORB を ORB リスト 204 から外すとともに、その ORB に対応するコマンドを I/O 要求キューから外す。

【0067】イニシエータたるホストコンピュータは以上のような機能的な構成からなっている。

【0068】＜ターゲットの構成＞図 1 は、ターゲットたるプリンタの機能上の構成を示すブロック図である。図 1 において、ドアベルレジスタ 101a は、イニシエータによって書き込まれるレジスタであり、ORB が新たに生成されたことを示す。また、ORB ポインタ 101b には、新たに生成された ORB のアドレスがイニシエータにより書き込まれる。フェッチエージェント 101 は、ドアベル 101a に書き込みがあると、1394 インターフェース 109 を介して ORB ポインタで示される ORB を読み込む。ORB ディスパッチャ 102 は、空参照リスト 104 からポインタを取り出し、ポインタの指し示すプリフェッチブル 103 の空領域にフェッチエージェント 101 により読み込まれた ORB のコマンドを格納するとともに、当該 ORB の後述する Queue ID フィールドにしたがって該当参照キューの末尾にポインタを接続する。

【0069】図中の例は、4 つの参照キュー 105a、105b、105c、105d が存在する状態を示している。参照キューの数はこれに限られるものではなく、参照キューは各クライアントプロセスの通信接続時に動的に用意され、各クライアントプロセスの通信には単数或いは複数の参照キューが用いられる。

【0070】実行エージェント 106 は、各参照キューにキューイングされたコマンドを先頭から取り出し、コマンドに従ってイニシエータのバッファへのデータの書き込み及びイニシエータのバッファからのデータの読み出し等を行う。その後、ホストコンピュータにノーマルステータスを返す。

【0071】実行エージェント 106 は 1 つの実行スレッドがすべての参照キューに対する処理を他の参照キューの処理にブロックされないように独立にスケジューリングして実行するようにしてもよいし、各参照キュー毎に別の実行スレッドでスレッドをスケジューリングし実行してもよいし、またその組み合わせであってもよい。なお、実行エージェント 106 は、例えば PDL を解釈実行してラスタデータを生成するラスタライザ等のクライアントプロセス 108a からのデータ転送要求に応じて、参照キュー 105b で参照されるプリフェッチブル内のリードコマンドで指定されたイニシエータのバッファにデータを書き込む。このデータ転送要求は、イニシエータからのライトコマンドやリードコマンドとは非同期に発生するため、イニシエータはリードコマンド ORB によって参照キュー 105b で参照されるプリフェッチブル内にリードコマンドを常時キューイングしておく。ターゲットは、リードコマンドが参照キュー 10

5 bで参照されるプリフェッチブール内にキューイングされている限り、いつでもデータをイニシエータに渡すことができる。なお、バスインターフェース110は、ターゲットであるプリンタ100からイニシエータであるホストコンピュータ200のシステムメモリ208の所望のメモリロケーションにアクセスするためのインターフェースである。以上、イニシエータ及びターゲットの構成と動作を簡単に説明した。これらを更に詳しく説明する前に、まずORBの詳しい内容を説明する。

【0072】<コマンドORB(Operation Request Block)の内容>図7はORB一般の構成を示す図である。

図7(A)において、"Next_ORB"(リンク)フィールド301は、次のORBへのリンクである。

【0073】次のORBがない場合にはそれを表す所定の値が入れられる。なお、先頭のORBは、所定のアドレスレジスタにより示される。

【0074】"data_descriptor"(データデスク립タ)フィールド302は、データバッファのアドレスを示す。"d"(方向)フィールド303は、ホストコンピュータからプリンタへのデータ転送(0:ライト)か、プリンタからホストコンピュータへのデータ転送(1:リード)かを示す。"data_size"(データサイズ)フィールド304は、アドレスフィールド302で指し示されるデータバッファのサイズを示す。以上、図の点線よりも上のフィールド(未説明のフィールドも含めて)がSBP-2で規定されているフィールドであり、これから説明するフィールド305~308が、SHP-2に固有の処理に用いられるフィールドである。

【0075】"QueueID"フィールド305は、使用される参照キューのIDを表わしている。"function"(ファンクション)フィールド306は、図7(B)に示すようにORBの種類を示している。0Hがライトコマンド、40Hがリードコマンドを表わす。

【0076】"seq_ID"(シーケンスID)フィールド308は、参照キュー毎に、ORBの生成順に振られるシーケンシャルな識別子である。この識別子は、本実施形態ではORBが生成されるごとに増加するように与えられる。

【0077】"seq_ID"フィールドは、最大値まで増加した後、次の増加で0に戻る。なお、このシーケンスIDフィールドに値を与える基準となるカウンタは、バスリセットや他の障害による影響を受けない領域に設けられる。それは、ターゲットにおいては、シーケンスIDの増加性を前提として処理を行うためである。コントロールブロックフィールド309には、ファンクションフィールド306の値や、各ファンクションに対してターゲットで実現される機能に応じて様々な値が入れられる。なお、バッファのアドレスやサイズについては、I/O要求キューに入れられたコマンドからORB生成時に得る。

【0078】次に、機能ごとにORBの内容を説明する。

【0079】(ライトコマンドORB)図8はファンクション=0HのライトコマンドORBを示している。

【0080】このコマンドは、指定したバッファ内のデータをイニシエータからターゲットに渡すためのコマンドである。ファンクションフィールドの値は0Hである。"i"(タグ)ビット307はデータタグを表している。

【0081】(リードコマンドORB)図9はファンクション=40HのリードコマンドORBを示している。

【0082】このコマンドは、イニシエータがターゲットからデータを読み出すためのコマンドである。ファンクションフィールドの値は40Hである。

【0083】(コネクトコマンドORB)図10はファンクション=C1HのコネクトコマンドORBを示している。このコマンドは、"QueueID"が0のコネクションサービス用参照キューに対して発行され、新たな接続を開き、その接続に用いる参照キューのIDを取得する為に用いられる。"Service_ID"フィールド309aには新たな接続先のサービスのIDを指定する。"Mode"フィールド309bには接続が全二重双方向、片方向、半二重双方向のいずれかを指定する。

【0084】"i"ビット309eはイニシエータがターゲットにプリフェッチブールの増加を要求するとき1にする。

【0085】(ディスコネクトコマンドORB)図11はファンクション=C2HのディスコネクトコマンドORBを示している。このコマンドは前記コネクトコマンドORBにより、開かれた接続を閉じるのに用いる。このコマンドは閉じようとしている参照キューに対して発行しても良いし、"QueueID"が0のコネクションサービス用参照キューに対して発行してもよい。"QueueID"が0のコネクションサービス用参照キューに対して発行した場合には、"QueueID1","QueueID2"フィールド309c,309dで指定された参照キューに対する接続が閉じる。接続に1つの参照キューのみを使用している場合は、"QueueID2"フィールドには0を指定する。

【0086】(クエリーコマンドORB)図12はファンクション=C0HのクエリーコマンドORBを示している。このコマンドはログイン後通信に先立ってターゲットからプリフェッチブールの容量を獲得するのに用いる。

【0087】(ステータスブロック)図13は、ターゲットからイニシエータに返されるステータスの形式及びその内容を示している。図13において、点線よりも上のフィールドはSBP-2で規定されたフィールドであるため、特に説明をしない。タグフィールド404はデータタグを表しており、リードコマンドORBに対するステータスでのみ有効である。

【0088】"SHPT2_status1"及び"SHPT2_status2" (SHPT-2ステータス) フィールドはコマンドORBの実行結果ステータスを示すフィールドで0以外でエラーを表わす。すなわち、値が0であれば、そのステータスに対応するコマンドORBについての処理は完了したことを示す。例えば、対応するコマンドORBがライトコマンドであれば、SHPT-2ステータスが0のステータスブロックは、対応するライトコマンドORBで差し示されたバッファから、ターゲットがデータを全て読み出し、ターゲットの持つバッファに書き込みを終えたことを意味する。また、値が0以外であれば、対応するコマンドORBの処理はエラーにより一部或いは全部が行われなかったことを示す。このエラーを受けたイニシエータは、クライアントにエラーである旨知らせるなどしてエラーに対応する。

【0089】"residual" (残余) フィールド406は、ORBで指し示されたバッファに対して処理されたデータ長を除いた残余データ長を示す。

【0090】(コネクタステータスブロック) 図14は、ファンクション=C1HのコネクタコマンドORBに対するステータスブロックであり、コネクタコマンド終了時に対してターゲットからイニシエータに渡される。新規接続に用いる参照キューのIDは"QueueID1"フィールド407a及び"QueueID2"フィールド407bに示される。コネクタコマンドORBの"Mode"フィールドが片方向の場合は"QueueID2"には0が返され、半二重の場合には"QueueID1"、"QueueID2"は同じIDが返され、双方向の場合には"QueueID1"、"QueueID2"には異なるIDが返される。"POOL_Inc"フィールド407cはターゲットのプリフェッチブールの容量増加分を示す。

【0091】(クエリーステータスブロック) 図15は、ファンクション=C0HのクエリーコマンドORBに対するステータスブロックであり、クエリーコマンド終了時に対してターゲットからイニシエータに渡される。"Init_POOL"フィールド407dにはターゲットのプリフェッチブールの容量が返される。

【0092】<ORBの管理>次に、以上説明したイニシエータ及びターゲットの構成や、コマンドORB及びステータスブロックの構成に基づいて、ORBがどのように用いられているか説明する。

【0093】図1、図2を用いて、イニシエータ、ターゲット及び各クライアント間におけるコマンドの流れを説明する。ここでは、要求キューA202a及び参照キューA105aがイニシエータ側からターゲット側へのデータ転送に使われ、要求キューB202b及び参照キューB105bがターゲット側からイニシエータ側へのデータ転送に使われるとして説明する。

【0094】イニシエータのクライアントからデータ転送要求が発生すると、クライアントはターゲットに渡すデータを格納したバッファのアドレス及びサイズを含む

ライトコマンドを、I/O要求キューA202aの末尾につける。

【0095】SHPT-2処理部201では、Current-QUEカウンタ203aが示すそのI/O要求キューに属するコマンドの現在のORBリスト内での個数がI/O要求キューに割り当てられたプリフェッチブール内における予約域の数以上かどうかをチェックし、予約域の数以下であれば、そのライトコマンドからORBを生成してORBリスト204の末尾にリンクし、Current-QUEカウンタ203aが示すそのI/O要求キューに属するコマンドの現在のORBリスト内での個数を1増やす。ORBリスト内での個数が予約域の数以上であれば、Current-POOLカウンタ203eをチェックし空がある、すなわち0でなければそのライトコマンドからORBを生成してORBリスト204の末尾にリンクし、Current-QUEカウンタ203aが示すそのI/O要求キューに属するコマンドの現在のORBリスト内での個数を1増やすとともにCurrent-POOLカウンタを1減ずる。

【0096】Current-POOLカウンタ203eをチェックし空がなければ空を待つ。図2の状態では、要求キューAから既に3つのORBがリンクされていて、予約域1以上になっているが、Current-POOLカウンタは3であるから、Current-POOLカウンタを2に減じ、Current-QUEカウンタのORBリスト内での個数を4に増やして、要求キューで待っていたライト要求Qa-4に対するORBをORBリストに追加できる。なお、ORBが生成されてもそれに対応するライトコマンドはI/O要求キューから削除されず、ターゲットからの完了ステータスを受けた時点で削除される。

【0097】SHPT処理部201は、ORBをORBリストに入れると、1394インターフェースを介してターゲットのドアベル101aに書き込む、あるいは、ORBポインタ101bに新たなORBのアドレスを書き込む。

【0098】ディスパッチャ102は、空参照リスト104からポインタを取り出し、ポインタの指し示すプリフェッチブール103の空領域にそのORBのコマンドを格納し、当該ORBのQueueIDフィールド(この場合は参照キューA105aを示している)にしたがって参照キューの末尾にポインタを接続する。実行エージェンツ106は、参照キューで参照されるプリフェッチブール内のコマンドを順次実行する。すなわち、ターゲットのクライアントの用意したバッファに、ライトコマンドで示されるバッファの内容を書き込む。実行エージェンツは、コマンドの処理が完了すると、SHPT-2ステータスが「完了」であるステータスブロックをイニシエータのステータスFIFO205に書き込む。

【0099】SHPT-2処理部201は、ステータスFIFOに入れられたステータスブロックを先頭から処理する。すなわち、SHPT-2ステータスが「完了」

であれば、そのステータスブロックに対応するORBをORBリストから外し、それとともにI/O要求キュー内の対応するライトコマンドを削除する。この時、Current-QUEカウンタ203aが示すそのI/O要求キューに属するコマンドの現在のORBリスト内での個数がI/O要求キューに割り当てられたプリフェッチプール内における予約域の数を超えているかどうかをチェックし、超えていなければ、Current-QUEカウンタ203aが示す現在のORBリスト内での個数を1減ずる。ORBリスト内での個数が予約域の数を超えている場合には、Current-POOLカウンタ203eを1増やし、Current-QUEカウンタ203aが示す現在のORBリスト内での個数を1減ずる。

【0100】以上の手順は、リードコマンドについても全く同様である。ただし、リードコマンドは、ターゲットにおいてリードすべきデータが発生しない限り処理されない。従って、イニシエータから発行されたリードコマンドORBは、リードすべきデータが発生するまでプリフェッチプール103に格納され参照キューB105bにキューイングされたままということになる。

【0101】イニシエータがホストコンピュータ、ターゲットがプリンタである場合、ホストコンピュータからプリンタに印刷データを渡すためには、上述のライトコマンドを用いて、イニシエータのクライアントであるプリンタドライバから、ターゲットのクライアントであるラスタライザに印刷データが渡される。また、ホストコンピュータがプリンタの構成や状態を示す情報を要求する場合には、その旨のコマンド（クライアントレベルのコマンド）をプリンタにライトコマンドで渡す。それを受けたプリンタ（のクライアント）は、プリフェッチプールに格納され、参照キューにキューイングされているリードコマンドを用いて、ホストコンピュータに要求されたデータを渡す。さらに、プリンタでエラーが発生したような場合には、プリンタのクライアントは、プリフェッチプールに格納され、参照キューにキューイングされているリードコマンドを用いて、自発的にエラー情報をホストコンピュータに渡すことができる。このため、ホストコンピュータはプリンタと接続している間は、運用上少なくとも1つのリードORBをプリンタに対して発行しておく。さらに、リードコマンドを常時プリフェッチプールに格納させ、参照キューにキューイングさせておくためには、少なくとも2つのリードORBをプリンタに対して発行しておくことが望ましい。

【0102】図2のORBリスト204は必ずしも先頭から順にORBが削除されていくとは限らない。例えば、参照キューB105bに向けられたORBT4（コマンドQb-1）は参照キューA105aに向けられたORBT3（コマンドQa-2）よりも先に完了し削除されることもある。このため、ORBT3（コマンドQa-2）のリンク先はなくなる。この場合、ORBT3

（コマンドQa-2）及びORBT5（コマンドQa-3）へのポインタはORBT4（コマンドQb-1）処理時にターゲットに既に渡っているため、ORBリスト上でリンク先をつなぎなおす必用はない。また、エラー等でORBリストがクリアされても、I/O要求キューには対応コマンドが残っているため、後述するエラー復旧処理も正常に行うことができる。

【0103】＜エラーの復旧（イニシエータ）＞既に説明したように、SBP-2では、ネットワークに異常（エラー）に起因するバスリセットにより、イニシエータとターゲットとの間の接続は断たれてしまう。そこで、SHP-2では、バスリセットにより失われた状態を復旧するための手順を定めている。

【0104】イニシエータはネットワーク上のエラーなどからバスリセットが発生後、バスリセットの処理が終了したなら、SBP-2の規定に従い全てのORBをORBリストから消去する。各I/O要求キューは、バスリセットがあってもリセットされることはない。

【0105】イニシエータは、コネクションを再設定した後、I/O要求キューを参照して、それらキューの内容から新たにORBを生成し、ターゲットに対して発行する。このとき、ORBの生成時に対応するコマンドに対して割り当てられていた“Seq_ID”フィールドの値を再度指定することにより、バスリセット直前の状態を再現することができる。

【0106】＜エラーの復旧（ターゲット）＞イニシエータはORBリストを復旧させるだけでその状態をバスリセット直前の状態に戻せるが、ターゲットはリード／ライトのコマンドに応じてバッファに対してリード／ライトを行うため、リード／ライト中にバスリセットが発生した場合には、復旧後はその続きを行わねばならない。そのために、ターゲットの実行エージェントは、各参照キューで参照されるそれぞれの処理中のコマンドの“Seq_ID”フィールドと、その処理中のバッファの位置を各参照キュー毎に記憶している。

【0107】例えば、イニシエータがリードORBを発行し、ターゲットはそれを実行エージェントによって処理する場合、実行前に、実行エージェント106はこれから処理するリードコマンドORBのseq_IDフィールドを読み出し、シーケンス識別子(Sequence identifier)として記憶しておく。また、クライアントのバッファにデータを書き込むに当たり、現在書き込んでいるバッファのアドレスを実行ポインタ(Next execution pointer)として更新しつづける。これらの領域はバスリセットによって消去されないメモリ領域に確保される。

【0108】バスリセットが終了したなら、イニシエータはORBを改めて発行してくるため、実行エージェントは、プリフェッチプールに格納され、参照キューの先頭で参照されるORBのseq_IDフィールドが、記憶されているシーケンス識別子よりも古ければ、そのORBは

既に処理は完了しているために、完了ステータスを返す。両者が一致すれば、Nextexcepointerで示されるアドレスから書き込みを続行する。

【0109】ライトコマンドについても同じ要領で復旧される。

【0110】以上、本実施形態の印刷システムにおいて用いられるコマンドおよびステータスについて説明した。次に、イニシエータおよびターゲットにおけるコマンド及びステータスの処理手順を説明する。

【0111】＜イニシエータのクライアントによるデータ転送要求＞図3は、イニシエータのクライアントであるプリンタドライバ等から、ターゲットに対して、データを送信あるいは要求する際の手順を示している。

【0112】データ転送事項が発生したなら（ステップS1301）、そのために必要なコマンドがライトであるかリードであるか判定し（ステップS1302）、ライトならばそのためのデータ転送バッファがあるか調べ（ステップS1303）、また、ターゲットに渡すべき送信データの準備ができていないか調べて（ステップS1304）、すべて準備できていれば、バッファのアドレスやサイズ等、必要な引き数を与えてライトコマンドを作成し（ステップS1305）、ライト関数をコールする（ステップS1306）。

【0113】一方、リードコマンドであると判定されれば、データを受けるための転送バッファがあるか調べ（ステップS1307）、あればバッファのアドレスやサイズ等、必要な引き数を与えてリードコマンドを作成し（ステップS1308）、リード関数をコールする（ステップS1309）。

【0114】例えば、データ転送事項がプリンタに対する印刷データの送信ならば、PDLなど、クライアントレベルのコマンドやデータをバッファに用意し、ライトコマンドを作成することになるし、データ転送事項がプリンタからの状態の読み出しであれば、その旨を示すクライアントレベルのコマンドを用意してライトコマンドを作成するとともに、ターゲットからデータを受けるためのリードコマンドを作成する必要がある。また、ターゲットとの一連のデータ交換を行うに先立って、イニシエータのクライアントはリードコマンドを幾つか発行する。

【0115】図4は、イニシエータのクライアントが下位すなわちSHPT-2の層から処理終了通知を受けた際の処理手順である。まず、終了した処理がリードかライトか判定し（ステップS1401）、ライトであれば使用したデータバッファを解放する（ステップS1402）。一方、リードであれば、処理の終了はデータの受信の完了であるから、受信したデータに対応した処理を行い（ステップS1403）、データバッファを解放する（ステップS1404）。

【0116】＜イニシエータのSHPT処理部による処

理＞図5～図6は、イニシエータのSHPT-2処理部による処理手順である。

【0117】図5（A）及び（B）は、それぞれクライアントがコールするライト関数及びリード関数の内容を示す。ライト関数では、そのクライアントが接続要求を出しコネクした時にクライアントに対してSHPT-2部が与えたコネクハンドルが引数として渡され、そのコネクハンドルに該当するI/O要求キューのIDに基づき、そのI/O要求キューにライトコマンドを追加する。リード関数も同様に、そのクライアントが接続要求を出しコネクした時にクライアントに対してSHPT-2部が与えたコネクハンドルが引数として渡され、そのコネクハンドルに該当するI/O要求キューのIDに基づき、そのI/O要求キューにリードコマンドを追加する。それぞれのI/O要求キューにキューイングされたコマンドは、後にターゲット側でコネクコマンドのステータスで与えられた"QueueID1", "QueueID2"で示される参照キューへそれぞれ送られることになる。

【0118】（I/O要求キュー管理）図5（C）各I/O要求キューの管理手順を示す。これらの手順は、各I/O要求キューごとに単独のスレッドでスレッドスケジュールによって独立に行われてもよいし、すべてのI/O要求キューをひとつのスレッド内で管理が他のI/O要求キューの動作によってブロックされないように独立スケジュールすることで実行してもよいし、またこれらの組み合わせであってもよい。図5（c）は1つのI/O要求キューの管理を示すものである。I/O要求キューの管理は各I/O要求キュー毎に行う。

【0119】まず、I/O要求キュー内にORB化されていないコマンドがあるか調べる（ステップS1601）。あれば、このI/O要求キューに属するコマンドの現在のORBリスト内での個数を示すカウンタCurrent-QUEがこのI/O要求キューに割り当てられたプリフェッチプール内における予約域の数より小さいか判定する（ステップS1602）。小さければS1605に進み、大きいか等しければS1604に進む。S1604では、プリフェッチプールに対するCurrent-P00Lカウンタが0より大きければCurrent-P00Lカウンタを1減じる。次にS1605でCurrent-QUEの値に1を加え（ステップS1605）、I/O要求キューに格納されている、ORB化されていない先頭のコマンドを基に、シーケンスID(Seq_ID)やSHPT-2コマンド(Function)を付加して、ORBを作成する（ステップS1606）。

【0120】こうして作成したORBをORBリストにつけ（ステップS1607）、ターゲットのドアベルに書き込む、あるいは、ORBポインタにORBのアドレスを書き込む（ステップS1608）。このようにして、I/O要求キュー内のコマンドからORBが作成さ

れる。

【0121】（ステータスブロックに対する処理）図6は、ステータスブロックをターゲットから受けた際の処理手順である。ORBに対する処理完了のステータスブロックを受けると、対応するORBをORBリストから削除し（ステップS1701）、対応ORBに対するコマンドがキューイングされているI/O要求キューを選択する（ステップS1702）。選択したI/O要求キューに属するコマンドの現在のORBリスト内での個数を示すカウンタCurrent-QUEがこのI/O要求キューに割り当てられたプリフェッチプール内における予約域の数以下であるかを判定し、以下であるならばS1704に進む。そうでなければ、S1704に進む（S1703）。S1704では、プリフェッチプールに対するCurrent-POOLカウンタを1増やす。次にS1705で選択したI/O要求キューに属するコマンドの現在のORBリスト内での個数を示すカウンタCurrent-QUEを1減ずる。最後に、選択されたI/O要求キューに対応するクライアントに処理の終了を通知する（ステップS1706）。クライアントはこれを受けて、図4の処理を行うことになる。

【0122】（エラー復旧処理）SHPT-2処理部における通常の処理は以上のとおりである。次にバスリセット後などに復旧する手順を図18により説明する。バスリセットが生じてターゲットとのSBP-2コネクションが切断されてしまった場合、あらためてSBP-2コネクションの再設定から行わねばならない。

【0123】まず、新規のORBのリンク処理を中断し（ステップS1801）、ORBリストを破棄する（ステップS1802）。この後SBP-2コネクションの再接続命令を出して（ステップS1803）、コネクションの確立を調べ（ステップS1804）、コネクションが再度張られたなら、各I/O要求キューの先頭のコマンドから順に対応するORBを作成する（ステップS1805）。このとき、各ORBの"Seq_ID"フィールドは、そのコマンドに対してバスリセット前に付与されていたのと同じIDを付与する。作成されたORBはORBリストにリンクされ（ステップS1806）、ORBポインタにアドレスを書き込んで（ステップS1807）、ターゲットにORBの発生を知らせる。

【0124】一方、コネクションが再設定できないまま一定時間経過すると（ステップS1808）、各I/O要求キュー内のコマンドをすべて破棄し（ステップS1809）、異常である旨をクライアントに知らせて（ステップS1810）、処理を終える。以上の手順により、リセット後に、あらためてターゲットとのSBP-2コネクションを確立し、リセット直前のORBリストを復旧することができる。

【0125】＜ターゲットによる処理＞図19～20は、ORBを受けたターゲットによる処理手順である。

【0126】（フェッチエージェントによる処理）フェッチエージェントは1つのSBP-2コネクションに対して1つ用意される。フェッチエージェントはドアベルレジスタあるいはORBポインタに書き込みがあるとORBをフェッチする。まず、ドアベルレジスタあるいはORBポインタに書き込みが行われるまで待ち（ステップS1901）、書き込みがあると、ドアベルレジスタへの書き込みにより起動されたか判定し（ステップS1902）、そうでなく、ORBポインタへの書き込みにより起動されたのであればステップS1903に進む。一方、ドアベルへの書き込みがあったならば、ORBポインタで指されているORBの"Next_ORB"フィールドがNULL（後続ORB無し）かどうかをテストする（ステップS1907）。NULLであればステップS1901に戻る。

【0127】NULLでなければ、ステップS1908へと進み、現在のORBポインタで指されているORBのNext_ORBフィールドの値を、ORBポインタへと書き込み更新し（ステップS1908）、ステップS1903へ進む。

【0128】こうしてORBポインタに処理すべきORBのアドレスが得られたなら、そのアドレスからORBを読み出す（ステップS1903）。空参照リストからプリフェッチプール内の空領域を指し示すポインタを取り出し、読み出したコマンドORBをその空領域に格納する（ステップS1905）。

【0129】またそのORBの"QueueID"フィールドを参照してコマンドをキューすべき参照キューを選択し（ステップS1905）、選択した参照キューの末尾に空参照リストから取り出したポインタをキューする（ステップS1906）。

【0130】この後、キュー処理を終えたORBのNext_ORBフィールドを参照して、その内容がNULL、すなわちリンクされているORBがなければ、S1901に戻り次のORBがリンクされ、ドアベルあるいはORBポインタへ書き込みがあるまで待つ。一方、リンクされているORBがあれば、ORBポインタで指されているORBのNext_ORBフィールドの値を、ORBポインタに書き込み更新し（ステップS1908）、ORBリストにリンクされているORBを順次プリフェッチプールに格納していく。

【0131】このようにして、ORBリストのORBをプリフェッチプールに入れる。

【0132】また、ステップS1904～S1906の処理は、図1においてORBディスパッチャとして説明した処理である。

【0133】（実行エージェント）図20は各参照キューに対する実行エージェントの手順を示す。これらの手順は、各参照キューごとに単独のスレッドでスレッドスケジュールによって独立に行われてもよいし、すべての

参照キューをひとつのスレッド内で管理が他の参照キューの動作によってブロックされないように独立スケジュールすることで実行してもよいし、またこれらの組み合わせであってもよい。図20は1つの参照キューに対する実行エージェントの手順を示すものである。各参照キューに対する実行エージェントの手順は各参照キュー毎に行う。

【0134】実行エージェントは、まず参照キューにプリフェッチブル内のコマンドへのポインタがあるか調べ(ステップS2001)、ORBをプリフェッチブルから取り出す(ステップS2002)。これらは参照キューの先頭を指し示すポインタを用いることにより行う。ORBを読み出すと、そのORBのSeq_IDフィールドの値が、実行エージェントが保持する変数SequenceIdentifierよりも小さい値であるか判定する(ステップS2003)。ただし、Seq_ID及びSequenceIdentifierはともに有限の桁数である。そこで、Seq_ID及びSequenceIdentifierがともにnビットで表わされる場合、ステップS2003における比較では、 $(2^n - 1) < 0$ ($= 2^n$)と定めておく。

【0135】通常のシーケンスであれば、ひとつのORBの処理終了後、SequenceIdentifierには後述のように1加算される。また、SequenceIdentifierとORBのSeq_IDは同じ桁数であり、Seq_IDも1ずつ加算される。そのため、支障なく処理が進んでいる場合には、SequenceIdentifierとORBのSeq_IDとは、ステップS2003において一致しているはずである。したがって、正常に処理が進めば、処理はステップS2003からステップS2004に移行する。

【0136】そこでは、クライアントがバッファを用意してあるか調べ(ステップS2004)、用意できたなら、ORBのdata_descriptorの値と実行エージェントが保持するNextexecPointerに格納されたoffset値とを加えたアドレスから所定サイズのデータアクセスを行い、クライアントが用意したバッファとの間でデータをやりとりする(ステップS2005)。

【0137】新たにORBの処理を行う場合には、offsetは0であるから、ORBにより指し示されるバッファの先頭から読み出すことになる。また、途中まで読み出されているバッファを続きから読む場合には、offsetは、バッファ中における続きのアドレスを示しているので、これを加算することで続きから読み出しを続行できる。データをバッファに格納したなら、data_descriptorの値+offsetが次に読み出されるアドレスを指すように、NextexecPointerを更新する(ステップS2006)。このような読み書きを、ORBのdata_sizeで示されるサイズ分に達するまで行う(ステップS2007)。

【0138】こうして1つのORBの処理を終えると、SequenceIdentifierに1加算して更新し(ステップS2

008)、NextexecPointer(offset)を0に初期化する(ステップS2009)。

【0139】処理を終えたなら、処理完了を通知するステータスブロックを作成し(ステップS2010)、ステータスFIFOに書き込み(ステップS2011)、ターゲットのクライアントに処理完了を通知する(ステップS2012)。それと同時に当該コマンドを指していた参照キュー内のポインタを空参照リストに戻す。

【0140】ステップS2003において、ORBのSeq_IDがSequenceIdentifierよりも小さい場合は、バスリセットなどにより、イニシエータが図18の手順でORBリストを再生したような場合である。例えば、あるORBについて、ターゲットは処理を終え、SequenceIdentifierの値も更新していても、そのステータスブロックがイニシエータに届いていない時点でバスリセットが発生すると、イニシエータはそのORBもORBリストに入れてしまう。この場合に、ORBのSeq_IDよりも実行エージェントのSequenceIdentifierの方が大きくなる。この場合には、そのORBは処理を終えているため、ステップS2010以降によりステータスブロックをイニシエータに渡す。

【0141】＜イニシエータの接続手順＞図16はイニシエータ側のクライアントからのオープン要求に対する処理手順を示したものである。クライアントからのオープン要求を受けると、まず、接続先のターゲットとの間に既にSBP-2のコネクション(ログイン状態)があるか否かを確認する(ステップS2501)。既にSBP-2コネクションがあれば、ステップS2507へ進み、まだSBP-2コネクションがなければ、ターゲットに対してログイン要求(SBP-2コネクト要求)を発行する(ステップS2502)。ログインが成功したかどうかを確認し(S2503)、ログインに失敗した場合はステップS2513に進み戻り値にエラーコードをセットしてリターンし、ログインに成功した場合は、QueueIDが0のI/O要求キューを作成し、そこにクエリーコマンドを発行する(ステップS2504)。次にクエリーコマンドが成功したか否かを確認し(ステップS2505)、失敗ならステップS2512へ進み、ログアウト実行後S2513へと進む。クエリーコマンドが成功したら、クエリーコマンドに対するステータスブロックの"Init_POOL"を元にCurrent-POOLカウンタをセットする。この時既にQueueIDが0に対応するI/O要求キューは暗黙に形成されているので、QueueIDが0に対応するI/O要求キューの予約域として1以上をセットし、残りをCurrent-POOLカウンタにセットする(ステップS2506)。

【0142】さらに、ステップS2507へ進んでQueueIDが0に対応するI/O要求キューに対して所望のServiceID.Modeを指定し、コネクトコマンドをキューする(ステップS2507)。コネクト時、ターゲットに対

して、プリフェッチプールの増量を求める場合は“i”ビットを1にセットする。次にコネクトコマンドが成功したか否かを確認し（ステップS2508）、失敗であれば、ステップS2511で暗黙に用意されるQueueID0以外で既に他のI/O要求キューが存在しているかどうかを確認し、存在していなければターゲットからログアウトし（ステップS2512）、S2513に進み、存在していれば、ログアウトせずにS2513に進む。一方、コネクトコマンドが成功するとステータスブロックのQueueID1, QueueID2に対応して、新しいI/O要求キューを作成し、後にクライアントがこのコネクションを
10 特定する為のハンドルを対応付ける。また、ステータスブロックのPOOL_Incフィールドと現在のCurrent-POOLレジスタの合計から、各キューに対して1以上の予約域をセットし、残りの数でCurrent-POOLレジスタを更新する（ステップS2509）。最後に作成されたI/O要求キューに対応させたハンドルを戻り値にセットし（ステップS2510）呼び出し元のクライアントにリターンする。

【0143】＜ターゲットの接続手順＞図17は、図16のイニシエータ側のクライアントからのオープン要求に対する処理手順に対応するターゲット側の処理手順を示したものである。

【0144】まず、ターゲットはログイン要求が来るのを待つ（ステップS2601）。

【0145】ログイン要求が来るとターゲットはSBP-2に規定された動作であるイニシエータのIDの確認、ログインディスクリプタの作成等を行うとともに、少なくとも1つのエントリを持ったプリフェッチプールを作成し、プリフェッチプール内のそれぞれのエントリ
30 に対するポインタのリストである空参照リストを作成する。また、QueueIDが0の参照キューの為のポインタ等も用意する（ステップ2602）。これらの準備が整ったら、ターゲットはイニシエータにSBP-2で規定されているログインレスポンスを返す（ステップ2603）。次にターゲットはクエリーコマンドが来るのを待つ（ステップ2604）。クエリーコマンドが来たら、作成したプリフェッチプールのサイズを“Init_POOL”フィールドに入れてクエリーコマンドに対するステータスブロックを送出する（ステップS2605）。ここま
40 でSHP-2による通信の準備が出来たことになる。

【0146】以降2610以下のループはQueueID0の参照キューに基づく実行エージェントの処理手順となる。S2606でコマンドがコネクトコマンドか否かを確認する。コネクトコマンドでなければS2609でそのコマンドの処理を行い2610へ戻る。コネクトコマンドであれば、Modeフィールドで指定されたモードにしたが
って1つ或いは2つの参照キューの為のポインタ等を作成する。この時、コネクトコマンドの“i”ビットが1であれば、作成した参照キューの数以上の数のエントリを
50

プリフェッチキューに追加し（ステップS2607）、作成した参照キューのID及びプリフェッチキューに追加したエントリ数をコネクトコマンドに対するステータスブロックでイニシエータに返す（ステップS2608）。この後から作成された参照キューを用いてクライアント間の通信が可能となる。

【0147】＜イニシエータの接続終了手順＞図21はイニシエータ側のクライアントからのクローズ要求に対する処理手順を示したものである。クローズ要求があると、まず、ディスコネクトコマンドを発行する。該当するQueueIDのI/O要求キューに既にキューイングされているコマンドを終了してからディスコネクトしたい場合にはディスコネクトコマンドのQueueIDフィールドを当該QueueIDにし、キューイングされているコマンドにかかわらずディスコネクトしたい場合は、QueueID0にディスコネクトコマンドを発行する。この場合、QueueID1, QueueID2フィールドでディスコネクトするキューを指定する（ステップS2701）。次にディスコネクトコマンドが完了するのを待ち（ステップS2702）、完了後、もし現在のORBリストの中にディスコネクトしたキューに対するORBが残っていればそれらをSBP-2の規定する制御フィールドによって全て無効化し、コネクト時に作成されたI/O要求キューを開放するとともに、そのI/O要求キューのCurrent-QUEカウンタで示されていた現在のORBリスト内のORB数分Current-POOLカウンタを増やし、さらにコネクト時に受け取ったプリフェッチプールの増加分を減ずる（ステップS2703）。さらに、ステップS2704で暗黙に用意されるQueueID0以外で既に他のI/O要求キューが存在
40 しているかどうかを確認し、存在していなければターゲットからログアウトし（ステップS2705）、そして、コネクト時に割り当てたハンドルを開放し、リターンする。

【0148】＜ターゲットの接続終了手順＞図22はターゲットのディスコネクト処理手順を示したものである。この手順は図17のステップS2609に位置するものである。まず、ディスコネクトコマンドかどうかを判定し（ステップS2801）、ディスコネクトコマンドでなければさらに他のコマンドの処理（ステップ2807）を行う。ディスコネクトコマンドであれば、ディスコネクトの対象となる参照キューで参照されているコマンドでディスコネクトコマンド以外に実行されているコマンドがあればその実行を停止する（ステップS2802）。実行を停止した後、その参照キューで参照されているディスコネクト以外のコマンドに対するステータスブロックで未送出のものがあれば全て送出する（ステップS2803）。その後ディスコネクトコマンドに対するステータスブロックを返す（ステップS2804）。この時点で当該参照キューに残っている未処理のポインターを全て空参照リストへと戻す（ステップS2

805)。コネクト時にプリフェッチブールの増加を行って、いればその増加分を減少し、減少分を参照している空参照リスト内のポインタを削除する。さらに参照キューのポインタ等も削除し（ステップS2806）リターンする。

【0149】図7に示したように、ORBにはQueueIDというキュー識別子のフィールドを備えている。そのため、SHP T-2 処理部及びORBディスパッチャがキュー識別子を識別して、それぞれがキューごとに独立し実行すなわちORBの生成・処理が行われることにより、複数のキューによるマルチコネクション（チャンネル）を実現することができる。この場合、1つの機器であっても、そこに含まれる複数のクライアントそれぞれに1つのコネクション（チャンネル）を割り当てることができ、クライアントごとに非同期の通信を行わせることができる。このため、例えばデジタル複合機であれば、それが有するスキャナやプリンタそれぞれについてクライアントとなるアプリケーションを設けておけば、それと接続されたホストコンピュータからは、あたかもそれぞれの機能を独立した機器であるかのように使用することができる。フローチャートで示したように、各キューの管理や実行エージェントは論理的に独立したプロセスであるために、マルチコネクション（チャンネル）化は容易に行うことができる。

【0150】またこのQueueIDにより識別される2つのキューを1つのコネクションに用いることによって、イニシエータとターゲットとの間で、簡素な制御手順によりデータの交換が双方向に行える。すなわち、イニシエータは所望のときに所望のデータをターゲットに渡すことができる。また、ターゲットは、イニシエータから渡されたデータをターゲット自身の都合に合わせて読み出すことができる。また、ターゲットはイニシエータが準備してさええすれば、自発的であろうと、あるいはイニシエータからの要求であろうと、何時でもイニシエータにデータを渡すことができる。また、バスリセットが発生しても、バスリセット直前の状態からの処理の続行を保証することができる。

【0151】さらに、キューによる多重化路におけるターゲットのコマンドブール領域全体の割り当てをイニシエータが動的に行うことによって、通信効率及びターゲットのリソース利用効率が向上するとともに、コネクションに用いられるキュー数に応じてターゲットのコマンドブール領域の増減が行われることにより、ターゲットのリソース利用効率が向上する。

【0152】（他の実施形態）なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェイス機器、リーダ、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【0153】また、本発明の目的は、イニシエータとし

て図3～図6、図16、図18、図21の手順のプログラムコードを記録した記憶媒体をコンピュータに供給し、そのコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても達成される。またターゲットも、図17、図19～図20、図22の手順のプログラムコードを記録した記憶媒体をコンピュータに供給し、そのコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによって実現できる。

【0154】この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0155】プログラムコードを供給するための記憶媒体としては、例えば、フロッピディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMなどを用いることができる。

【0156】また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。

【0157】さらに、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。

【0158】また、記憶媒体から読み出されたプログラムコードを実行してイニシエータとなる主体はコンピュータに限らず、データ転送機器であって、SBP-2のログイン能力があれば、どのような機器であっても良い。

【0159】また、ターゲットは、複数の参照キューを用いることにより、たとえば、第1の参照キューに印刷ジョブを格納し、第2のキューにターゲットの状態を確認するためのリードコマンドを格納し、第3のキューにフィニッシャー制御コマンドを格納することにより、パイプライン処理が実現でき、印刷ジョブに基づく印刷が完了したら、フィニッシャーによる機能の実行が終了する前に、第1の参照キューに次の印刷ジョブを格納できる。

【0160】（第2の実施の形態）第一の実施の形態においては、ディスコネクト時プリフェッチブールの容量

は、対応するコネク特時の増量分だけ減少するように構成されているが、ディスコネク特時にイニシエータが減少分を指示するように構成されてもよい。この場合、図 11 のディスコネク特コマンド ORB に対して、図 23 の如く、POOL_Dec フィールド 309 f を設け、イニシエータがディスコネク特コマンドを発行時にこのフィールドにターゲットに開放させるブリフェッチブールの量を指定し、図 21 のステップ S2701 でコマンドを発行し、ターゲットは図 22 のステップ S2806 で POOL_Dec フィールドで指定された量だけブリフェッチブールを減ずればよい。

【0161】この実施の形態によっても第一の実施の形態同様の下記のごとき効果が得られる。SHPT-2 処理部及び ORB ディスパッチャがキュー識別子を識別して、それぞれがキューごとに独立し実行すなわち ORB の生成・処理が行われることにより、複数のキューによるマルチコネクション（チャンネル）を実現することができる。この場合、1 つの機器であっても、そこに含まれる複数のクライアントそれぞれに 1 つのコネクション（チャンネル）を割り当てることで、クライアントごとに非同期の通信を行わせることができる。

【0162】このため、例えばデジタル複合機であれば、それが有するスキャナやプリンタそれぞれについてクライアントとなるアプリケーションを設けておけば、それと接続されたホストコンピュータからは、あたかもそれぞれの機能を独立した機器であるかのように使用することができる。フローチャートで示したように、各キューの管理や実行エージェントは論理的に独立したプロセスであるために、マルチコネクション（チャンネル）化は容易に行うことができる。

【0163】またこの QueueID により識別される 2 つのキューを 1 つのコネクションに用いることによって、イニシエータとターゲットとの間で、簡素な制御手順によりデータの交換が双方向に行える。すなわち、イニシエータは所望のときに所望のデータをターゲットに渡すことができる。また、ターゲットは、イニシエータから渡されたデータをターゲット自身の都合に合わせて読み出すことができる。また、ターゲットはイニシエータが準備していさえすれば、自発的であろうと、あるいはイニシエータからの要求であろうと、何時でもイニシエータにデータを渡すことができる。また、バスリセットが発生しても、バスリセット直前の状態からの処理の続行を保証することができる。

【0164】さらに、キューによる多重化路におけるターゲットのコマンドブール領域全体の割り当てをイニシエータが動的に行うことによって、通信効率及びターゲットのリソース利用効率が向上するとともに、コネクションに用いられるキュー数に応じてターゲットのコマンドブール領域の増減が行われることにより、ターゲットのリソース利用効率が向上する。

【0165】（第 3 の実施の形態）また、第一の実施の形態においてはイニシエータはコネク特コマンドにおいて "i" ビットにより、ブリフェッチブールの増加を要求するか否かのみを指定するように構成されていたが、イニシエータが増加数を指定して増加を要求することも可能である。この場合、図 10 のコネク特コマンドにおける "i" フィールドのビット数を増やし、増加数を示すようにすればよい。さらに、第一の実施の形態においては、イニシエータ側におけるキューに対するブリフェッチブールに対応した予約域はコネク特時にのみ設定されるように構成されていたが、コネク特時以外にも予約域を増減できるように構成することも可能である。この場合、予約域の増加分は Current-POOL カウンタの値を超えないようにし、その範囲内で、予約域数を増し、Current-POOL カウンタを減ずる。減少するには、Current-QUE カウンタで示される現在の ORB リスト内での当該キューに属する ORB 数が予約域数を下回っている場合、その差分を上限として、その範囲内でなおかつ減少後の予約域数が 1 以上となる範囲で減少をさせることが可能である。この場合、その範囲内で予約域数を減じ、Current-POOL カウンタを増やす。

【0166】この実施の形態によっても第一の実施の形態同様の下記のごとき効果が得られる。SHPT-2 処理部及び ORB ディスパッチャがキュー識別子を識別して、それぞれがキューごとに独立し実行すなわち ORB の生成・処理が行われることにより、複数のキューによるマルチコネクション（チャンネル）を実現することができる。この場合、1 つの機器であっても、そこに含まれる複数のクライアントそれぞれに 1 つのコネクション（チャンネル）を割り当てることで、クライアントごとに非同期の通信を行わせることができる。

【0167】このため、例えばデジタル複合機であれば、それが有するスキャナやプリンタそれぞれについてクライアントとなるアプリケーションを設けておけば、それと接続されたホストコンピュータからは、あたかもそれぞれの機能を独立した機器であるかのように使用することができる。フローチャートで示したように、各キューの管理や実行エージェントは論理的に独立したプロセスであるために、マルチコネクション（チャンネル）化は容易に行うことができる。

【0168】またこの QueueID により識別される 2 つのキューを 1 つのコネクションに用いることによって、イニシエータとターゲットとの間で、簡素な制御手順によりデータの交換が双方向に行える。すなわち、イニシエータは所望のときに所望のデータをターゲットに渡すことができる。また、ターゲットは、イニシエータから渡されたデータをターゲット自身の都合に合わせて読み出すことができる。また、ターゲットはイニシエータが準備していさえすれば、自発的であろうと、あるいはイニシエータからの要求であろうと、何時でもイニシエータ

にデータを渡すことができる。また、バスリセットが発生しても、バスリセット直前の状態からの処理の続行を保証することができる。

【0169】さらに、キューによる多重化路におけるターゲットのコマンドブール領域全体の割り当てをイニシエータが動的に行うことによって、通信効率及びターゲットのリソース利用効率が向上するとともに、コネクシオンに用いられるキュー数に応じてターゲットのコマンドブール領域の増減が行われることにより、ターゲットのリソース利用効率が向上する。

【0170】（第4の実施の形態）上記実施の形態においては、プリフェッチブールの増加・減少は、コネク（図10）・ディスコネクコマンド（図23）の“i”ビット309e, “POOL_Dec”フィールド等で指定していたか、コネクステータス（図14）の“POOL_Inc”フィールドで指定されたものを用いていたが、これらの代わりにプリフェッチブール増加・減少の為の独立したコマンドを設けてもよい。

【0171】図25はプリフェッチブール容量変更コマンド（ファンクションコード＝C3H）のフォーマットを示している。図25において“delta”フィールド309gは2の補数形式の符号付の数値を示しており、このフィールドが正であればプリフェッチブールの増加を、負であればプリフェッチブールの減少をターゲットに指示する。ターゲットはこのコマンドの指示に従って実際にプリフェッチブールの増加・減少を行う。

【0172】このコマンドを使った場合、イニシエータのクライアントからのオープン要求処理時には、図16のステップS2507に先だって、プリフェッチブールの増加が必要であればイニシエータはプリフェッチブール変更コマンドを発行し、プリフェッチブールを増加させる。ターゲットは図17におけるステップS2607以前に必要なであればすでにプリフェッチブール変更コマンドを別途受けているので、ステップS2607では、参照キューの為のポインタ等のみを作成することになる。また、イニシエータのクライアントからのクローズ要求処理時には、ディスコネクコマンドが成功した後（ステップS2703）に、プリフェッチブール容量変更命令を発行してプリフェッチブールを減少させる。ターゲットは後にイニシエータからプリフェッチブール容量変更コマンドが発行されるはずなので、ディスコネクコマンドの処理（図22）においては、ステップS2806では参照キューに用いるポインタ等のみを削除し、プリフェッチブールの減少は行わない。

【0173】この実施の形態によっても第一の実施の形態同様の下記のごとき効果が得られる。SHPT-2処理部及びORBディスパッチャがキュー識別子を識別して、それぞれがキューごとに独立し実行すなわちORBの生成・処理が行われることにより、複数のキューによるマルチコネクシオン（チャンネル）を実現することがで

きる。この場合、1つの機器であっても、そこに含まれる複数のクライアントそれぞれに1つのコネクシオン（チャンネル）を割り当てることで、クライアントごとに非同期の通信を行わせることができる。

【0174】このため、例えばデジタル複合機であれば、それが有するスキャナやプリンタそれぞれについてクライアントとなるアプリケーションを設けておけば、それと接続されたホストコンピュータからは、あたかもそれぞれの機能を独立した機器であるかのように使用することができる。フローチャートで示したように、各キューの管理や実行エージェントは論理的に独立したプロセスであるために、マルチコネクシオン（チャンネル）化は容易に行うことができる。

【0175】またこのQueueIDにより識別される2つのキューを1つのコネクシオンに用いることによって、イニシエータとターゲットとの間で、簡素な制御手順によりデータの交換が双方向に行える。すなわち、イニシエータは所望のときに所望のデータをターゲットに渡すことができる。また、ターゲットは、イニシエータから渡されたデータをターゲット自身の都合に合わせて読み出すことができる。また、ターゲットはイニシエータが準備していさえすれば、自発的であろうと、あるいはイニシエータからの要求であろうと、何時でもイニシエータにデータを渡すことができる。また、バスリセットが発生しても、バスリセット直前の状態からの処理の続行を保証することができる。

【0176】さらに、キューによる多重化路におけるターゲットのコマンドブール領域全体の割り当てをイニシエータが動的に行うことによって、通信効率及びターゲットのリソース利用効率が向上するとともに、コネクシオンに用いられるキュー数に応じてターゲットのコマンドブール領域の増減が行われることにより、ターゲットのリソース利用効率が向上する。

【0177】

【発明の効果】以上説明したように、本発明によれば、イニシエータとターゲットとの間において、1回のログインで非同期的な双方向通信及びそのマルチチャンネル化を可能とし、また、データの交換に必要なプロセスやメモリといった資源を効率的に利用できる。

【0178】また、IEEE1394インターフェースを利用しているために、ターゲット側へのデータ転送を、ターゲットがその都合に応じて読み出すことで行うことができ、イニシエータがターゲットの都合によってデータ転送に占有されることを防止できる。

【0179】また、SBP-2プロトコルを利用しているため、ターゲットでキューされるのはORBだけで、実際に転送されるデータそのものは処理待ちの間イニシエータに格納されている。このため、ターゲットのメモリ資源を小容量化できる。

【0180】また、最新の処理状態を保持することで、

バスリセットが発生しても、バスリセット後に、バスリセット直前の状態から処理を再開することができ、処理の正常な続行を保証することができる。

【0181】また、イニシエータからターゲットに送信できる命令の数の管理をターゲットの各キュー毎に行うのではなく、一元管理することにより効率的にリソースを利用できる。

【0182】また、キューによる多重化路におけるターゲットのコマンドブール領域全体の割り当てをイニシエータが動的に行うことによって、通信効率及びターゲットのリソース利用効率が向上するとともに、コネクションに用いられるキュー数に応じてターゲットのコマンドブール領域の増減が行われることにより、ターゲットのリソース利用効率が向上できる。

【図面の簡単な説明】

【図1】ターゲット（プリンタ）のブロック図である。

【図2】イニシエータ（ホストコンピュータ）のブロック図である。

【図3】イニシエータのクライアントによる、データ転送要求の発生時の処理手順のフローチャートである。

【図4】イニシエータのクライアントによる、SHPT-2からの処理終了通知受信時の処理手順のフローチャートである。

【図5】イニシエータのSHPT-2処理部により実行される、クライアントからコールされるライト関数、リード関数のフローチャート、及び、イニシエータのSHPT-2処理部により実行される、I/O要求キューの管理手順のフローチャートである。

【図6】イニシエータのSHPT-2処理部により実行される、ステータスブロック受信時の処理手順のフローチャートである。

【図7】ORB一般のフォーマットを示す図である。

【図8】ライトコマンドORBのフォーマットを示す図である。

【図9】リードコマンドORBのフォーマットを示す図である。

【図10】コネクトコマンドORBのフォーマットを示す図である。

【図11】ディスコネクトコマンドORBのフォーマットを示す図である。

【図12】クエリーコマンドORBのフォーマットを示す図である。

【図13】ステータスブロック一般のフォーマットを示す図である。

【図14】コネクトステータスブロックのフォーマットを示す図である。

【図15】クエリーステータスブロックのフォーマット

を示す図である。

【図16】イニシエータのSHPT-2処理部により実行される、クライアントからのオープン要求に対する処理手順のフローチャートである。

【図17】ターゲットのコネクト要求に対する処理手順のフローチャートである。

【図18】イニシエータのSHPT-2処理部により実行される、バスリセット直後の復旧処理手順のフローチャートである。

【図19】ドアベルレジスタあるいはORBポインタに書き込みが行われた際のターゲットのフェッチエージェントによる処理手順のフローチャートである。

【図20】ターゲットの実行エージェントによる処理手順のフローチャートである。

【図21】イニシエータのSHPT-2処理部により実行される、クライアントからのクローズ要求に対する処理手順のフローチャートである。

【図22】ターゲットのディスコネクト要求に対する処理手順のフローチャートである。

【図23】第2の実施の形態におけるディスコネクトコマンドORBのフォーマットを示す図である。

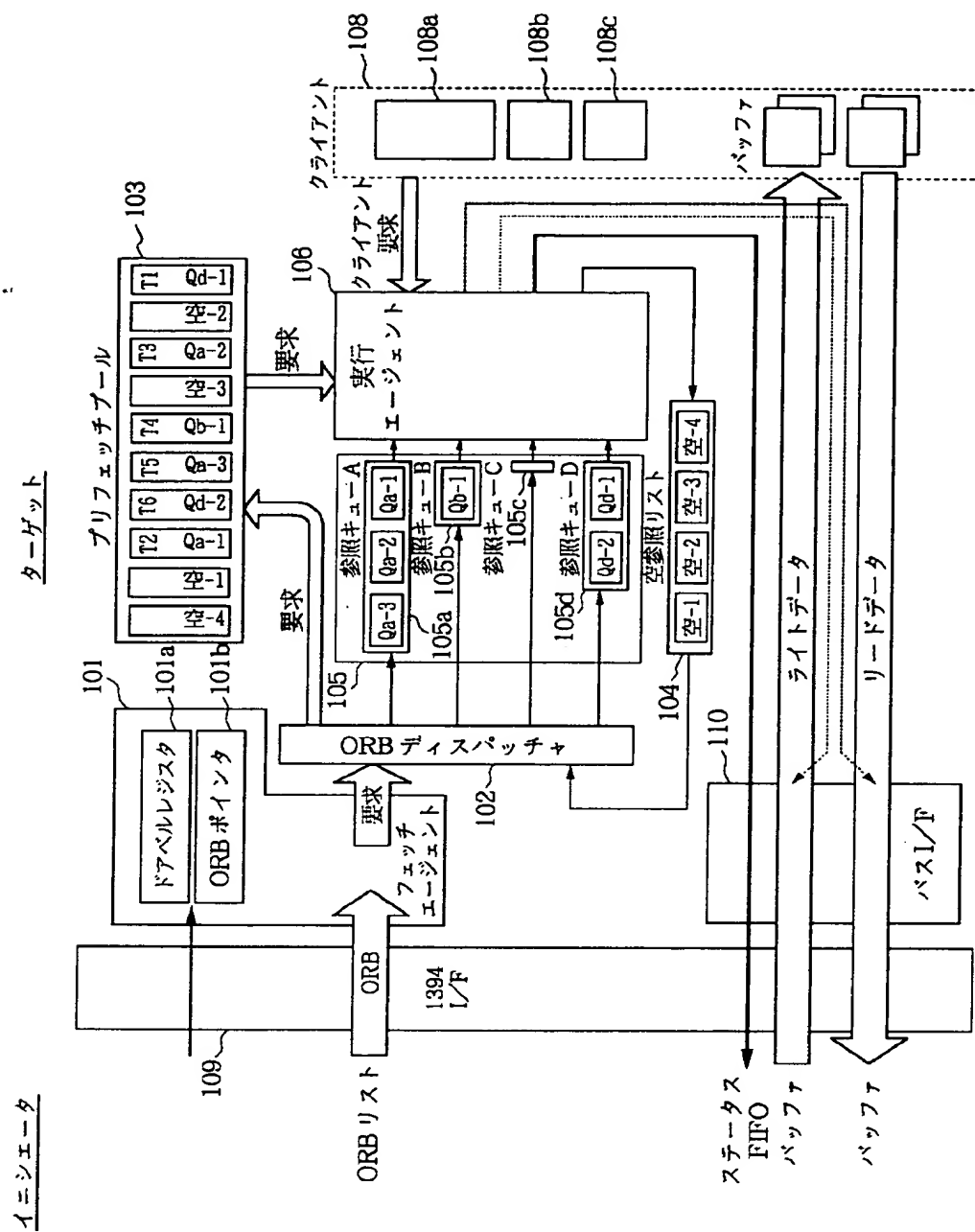
【図24】IEEE1394インターフェースを用いたプリンタシステムのハードウェア構成図である。

【図25】プリフェッチブール容量変更コマンドのフォーマットを示す図である。

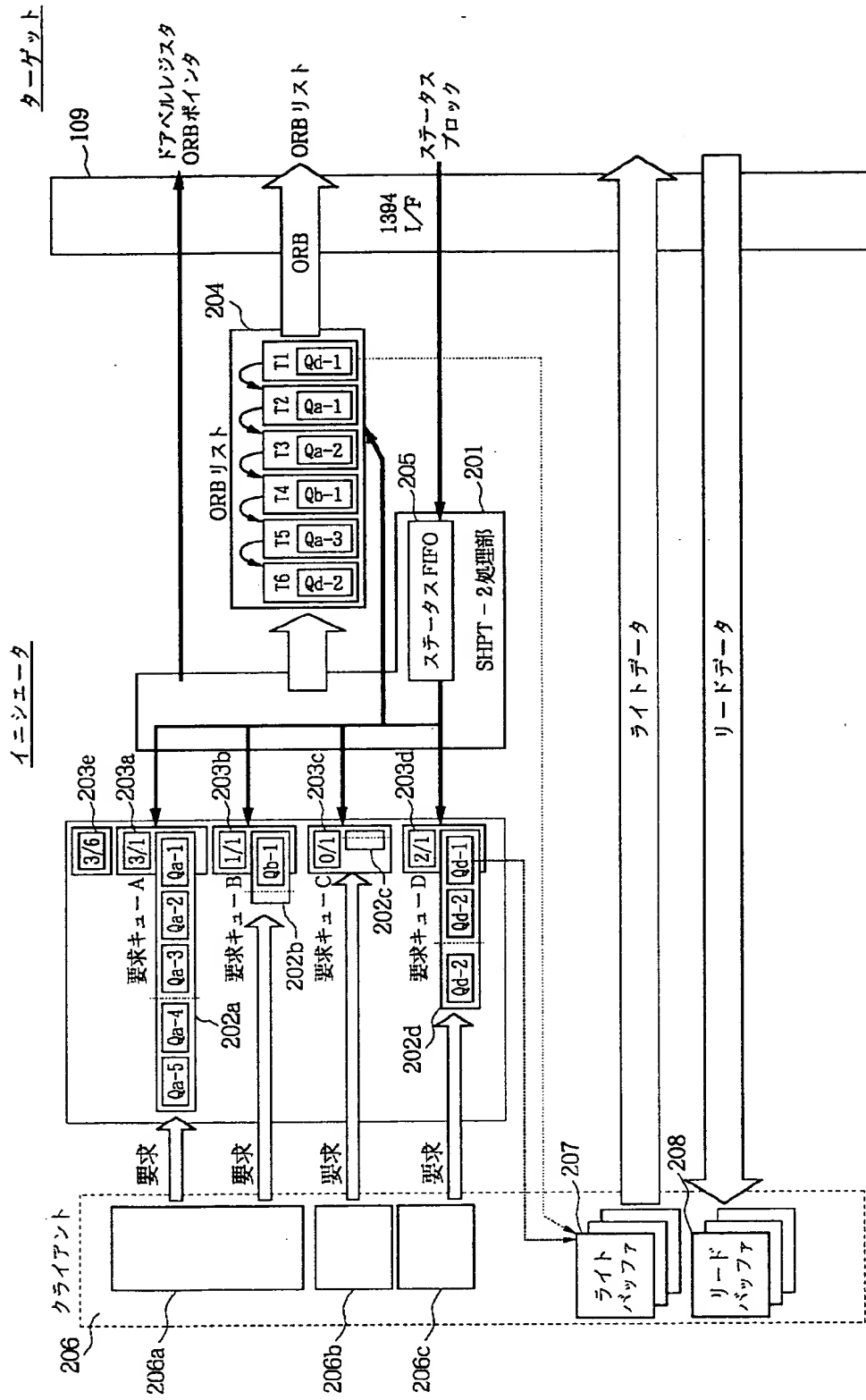
【符号の説明】

- 1 CPU
- 2 RAM
- 3 ROM
- 4 システムバス
- 12 CPU
- 13 ROM
- 19 RAM
- 101 フェッチエージェント
- 103 プリフェッチブール
- 105 参照キュー
- 106 実行エージェント
- 108 ターゲット側クライアント
- 109 1394インターフェース
- 110 バスインターフェース
- 201 SHPT-2処理部
- 202a I/O要求キュー
- 204 ORBリスト
- 205 ステータスFIFO
- 206 イニシエータ側クライアント
- 207 ライトバッファ
- 208 リードバッファ

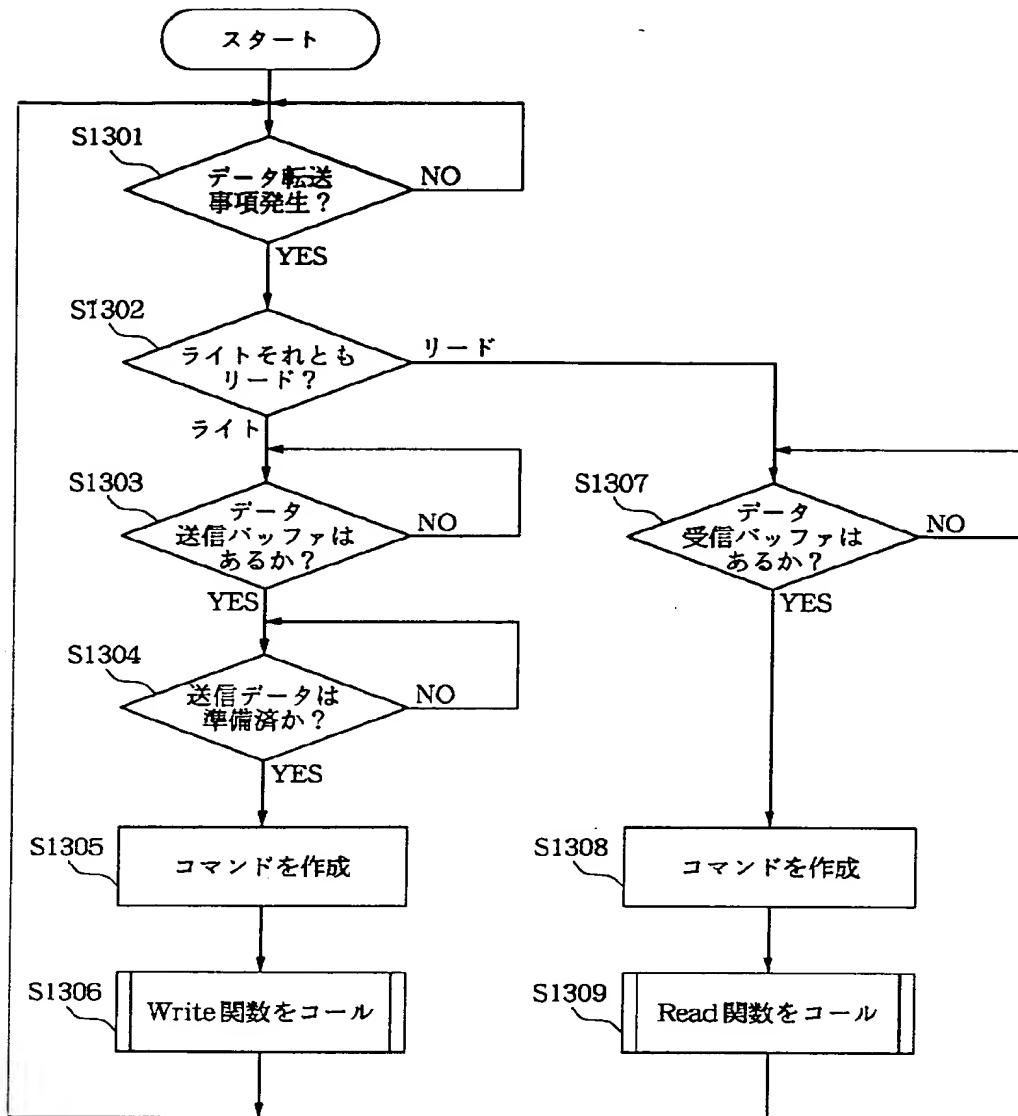
【图 1】



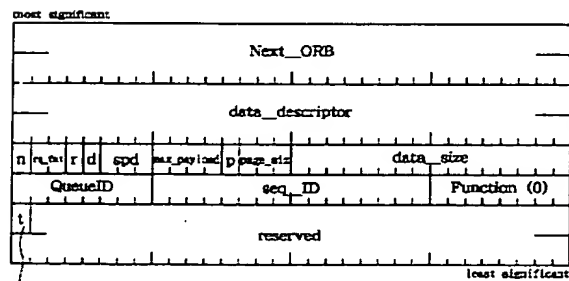
【図 2】



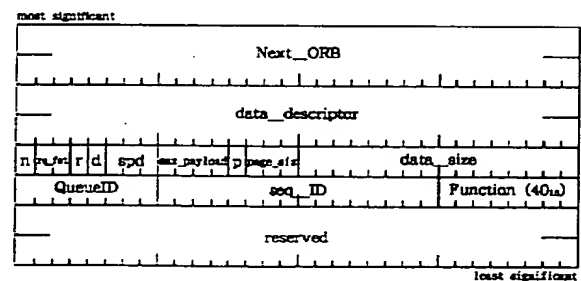
【図 3】



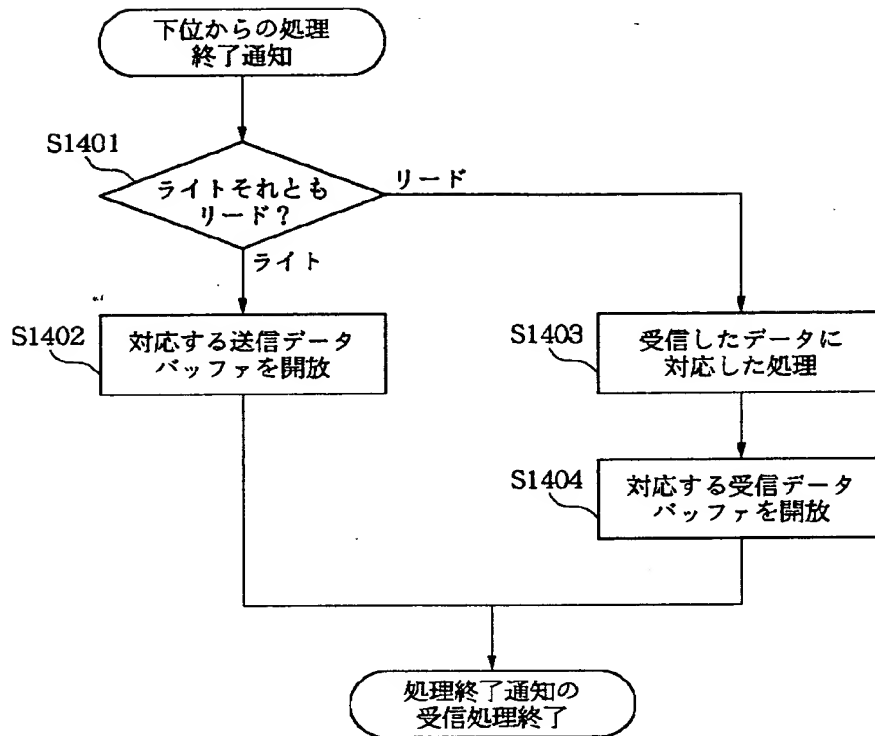
【図 8】



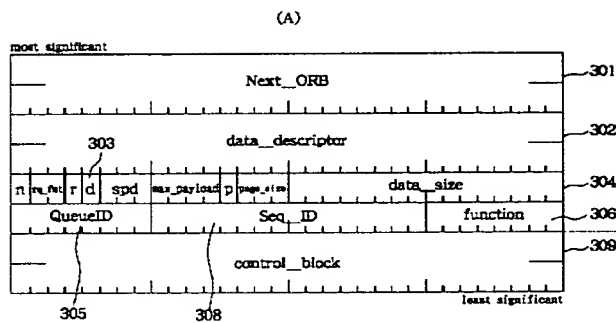
【図 9】



【図 4】



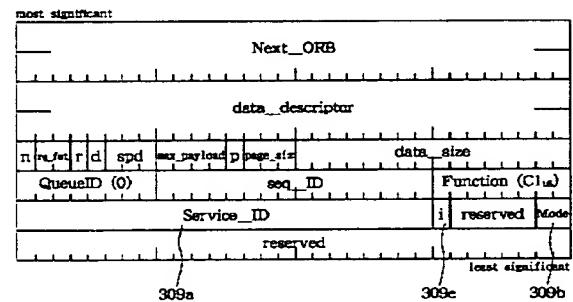
【図 7】



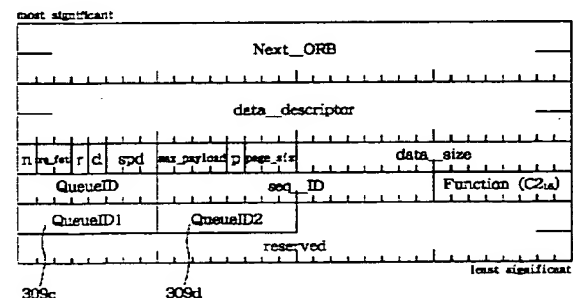
(B)

Value	SHPT-2 コマンド
0	ライト
1-3F ₁₆	reserved for future standardization
40 ₁₆	リード
41 ₁₆ -BF ₁₆	reserved for future standardization
80 ₁₆ -BF ₁₆	service dependent
C0 ₁₆	reserved for future standardization
C1 ₁₆	コネクト
C2 ₁₆	ディスコネクト
C3 ₁₆ -FF ₁₆	reserved for future standardization

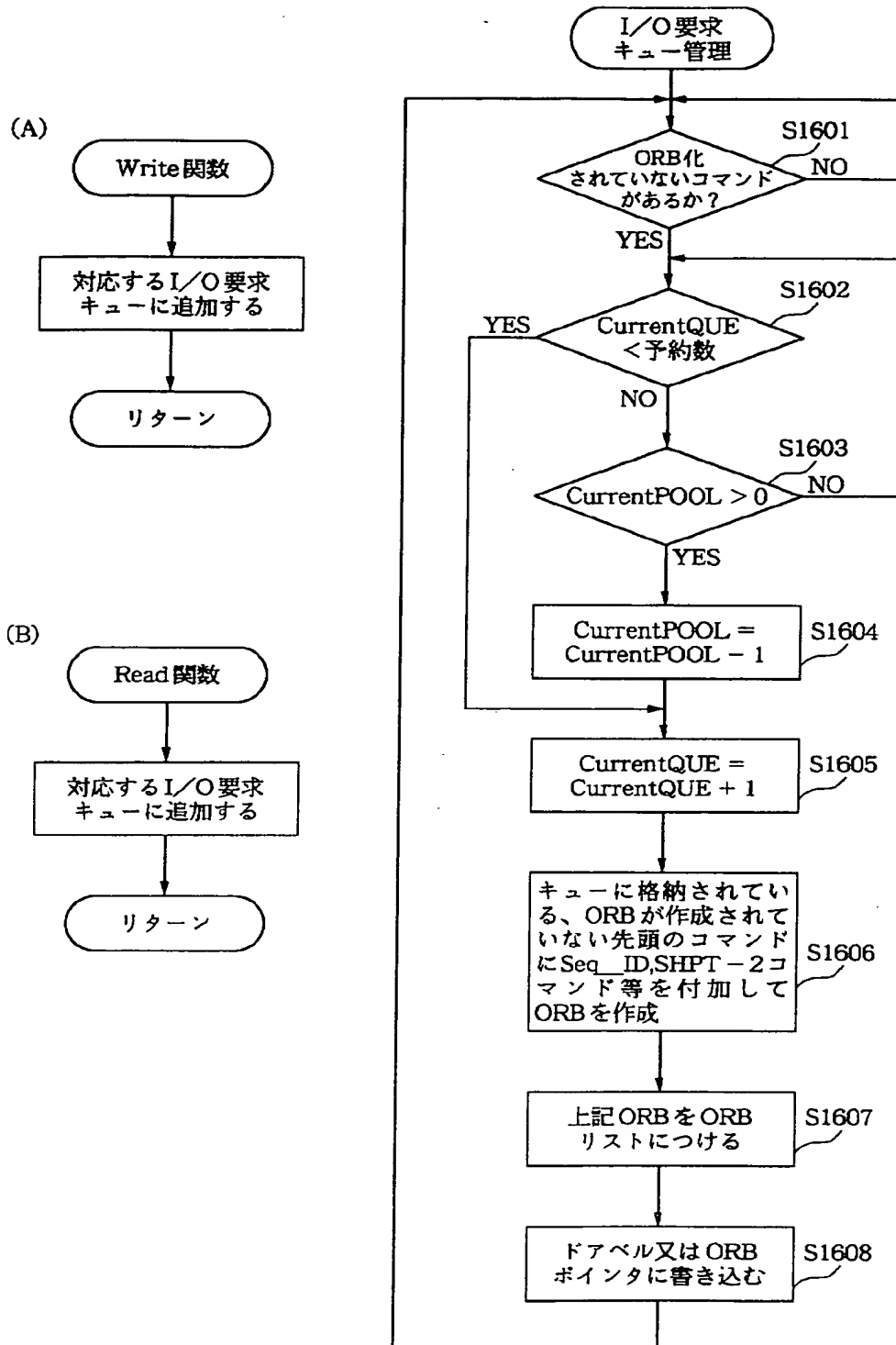
【図 10】



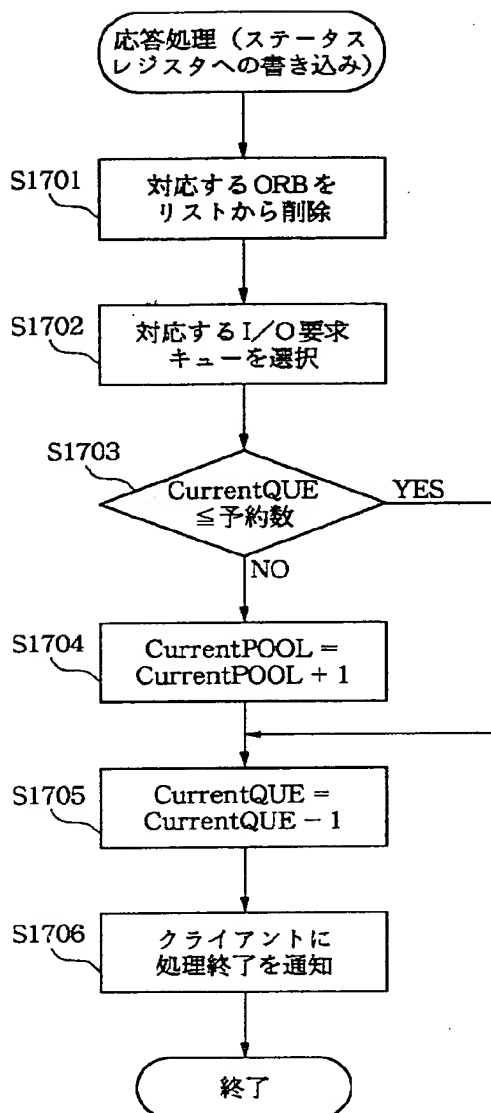
【図 11】



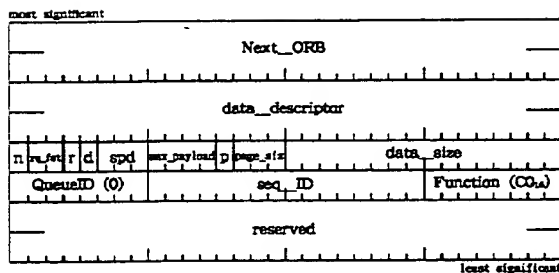
【図 5】



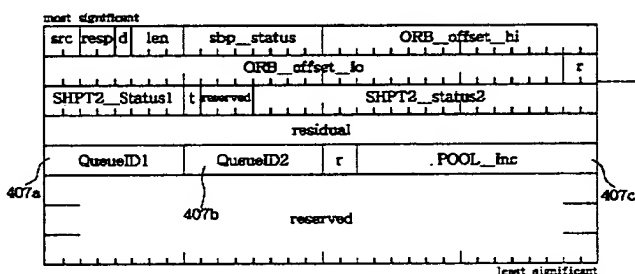
【図 6】



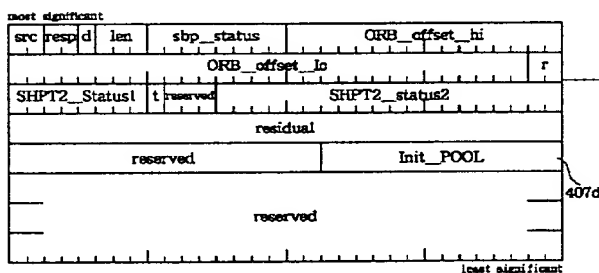
【図 12】



【図 14】

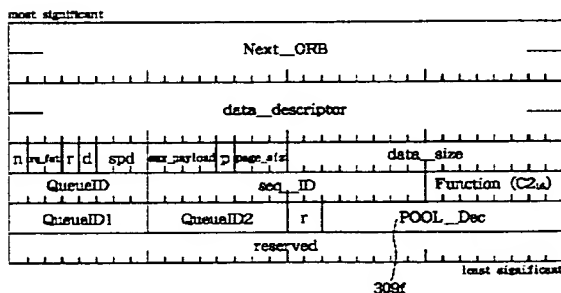
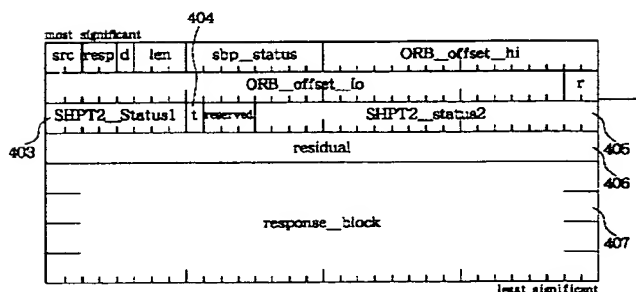


【図 15】

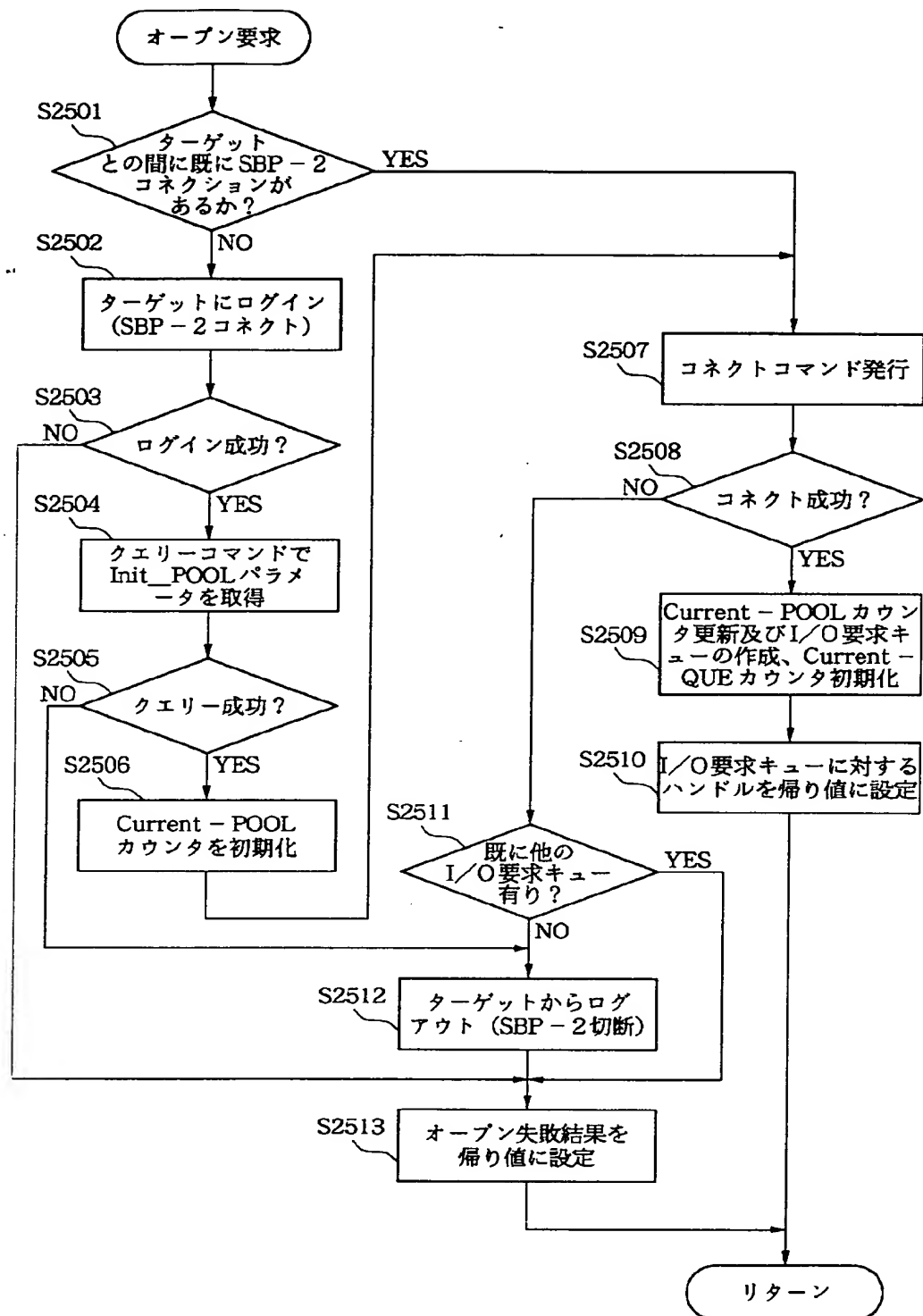


【図 23】

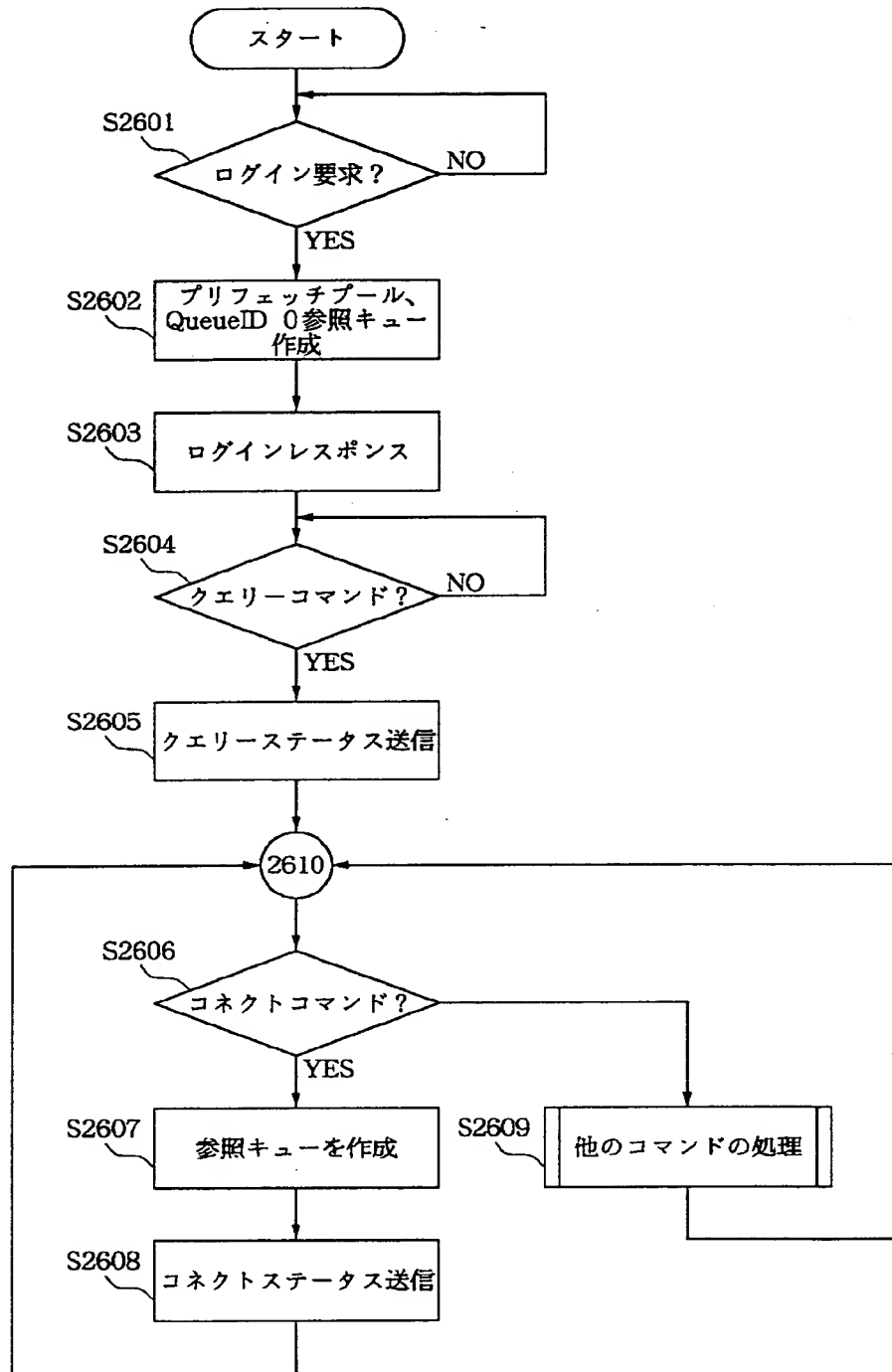
【図 13】



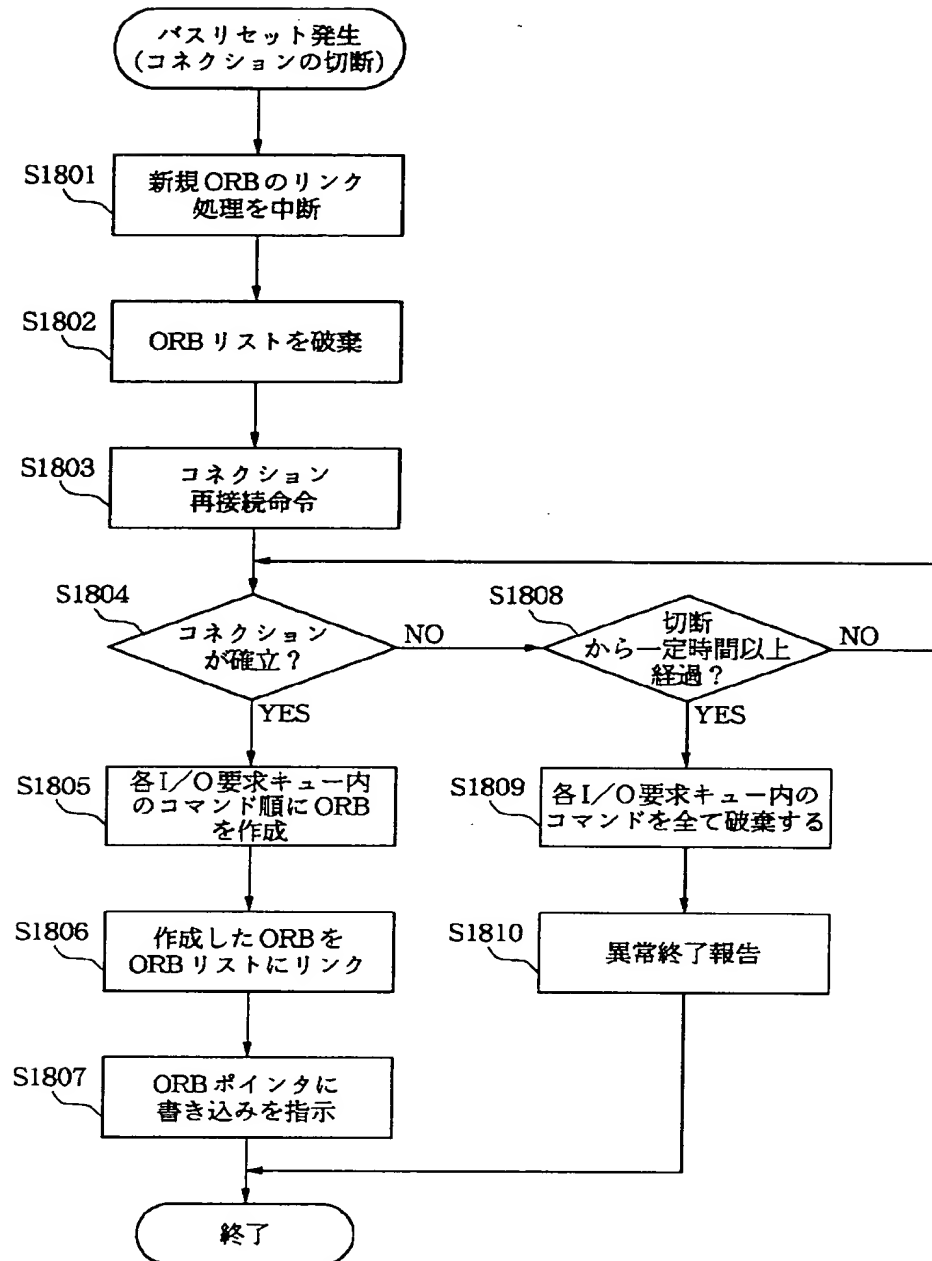
【図 16】



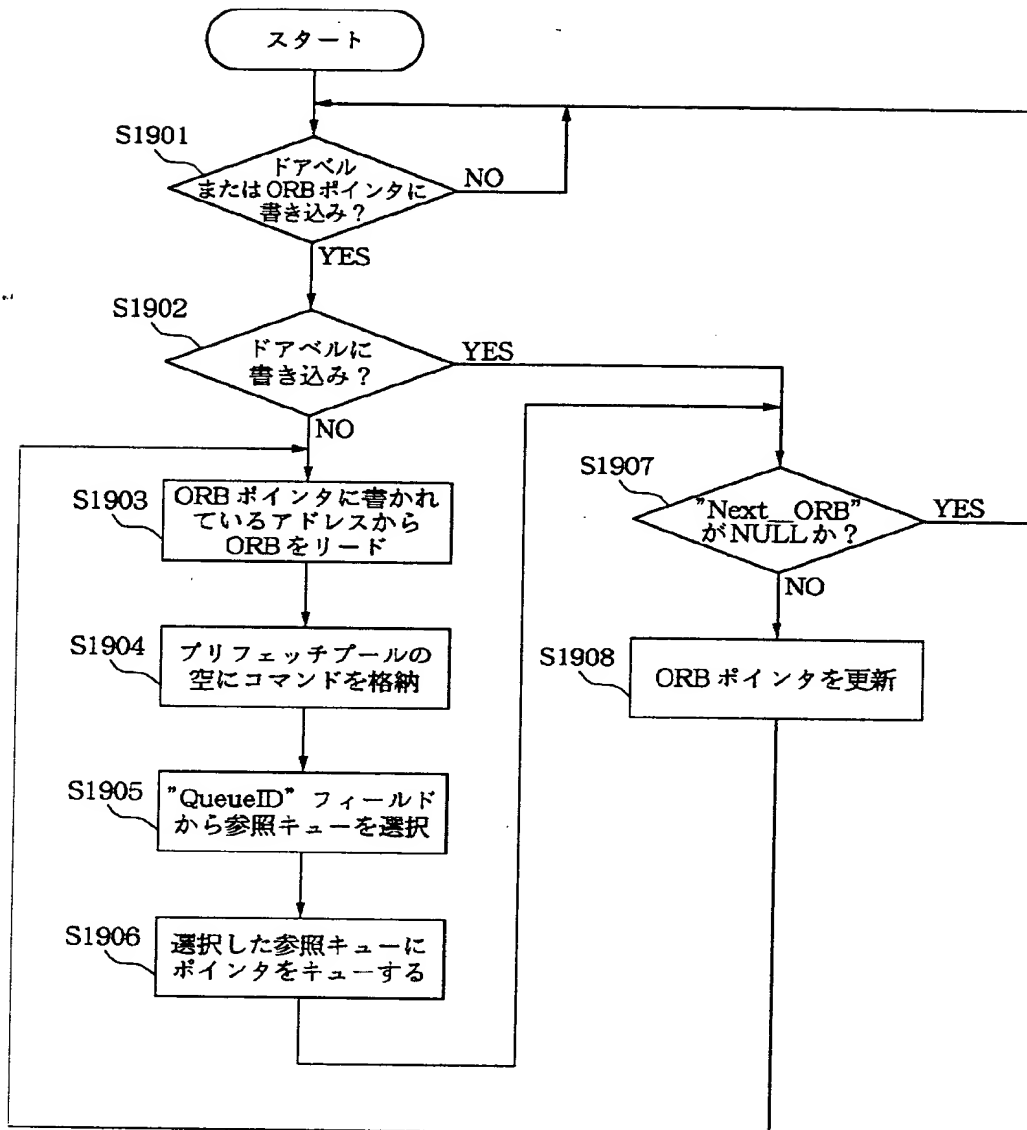
【図 17】



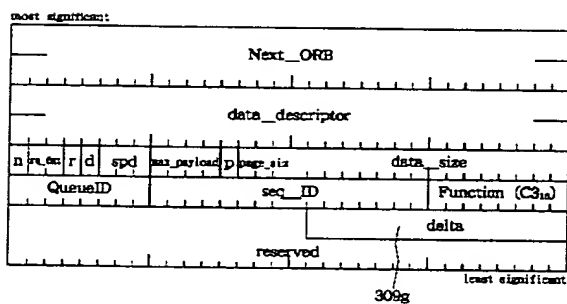
【図 18】



【図 19】

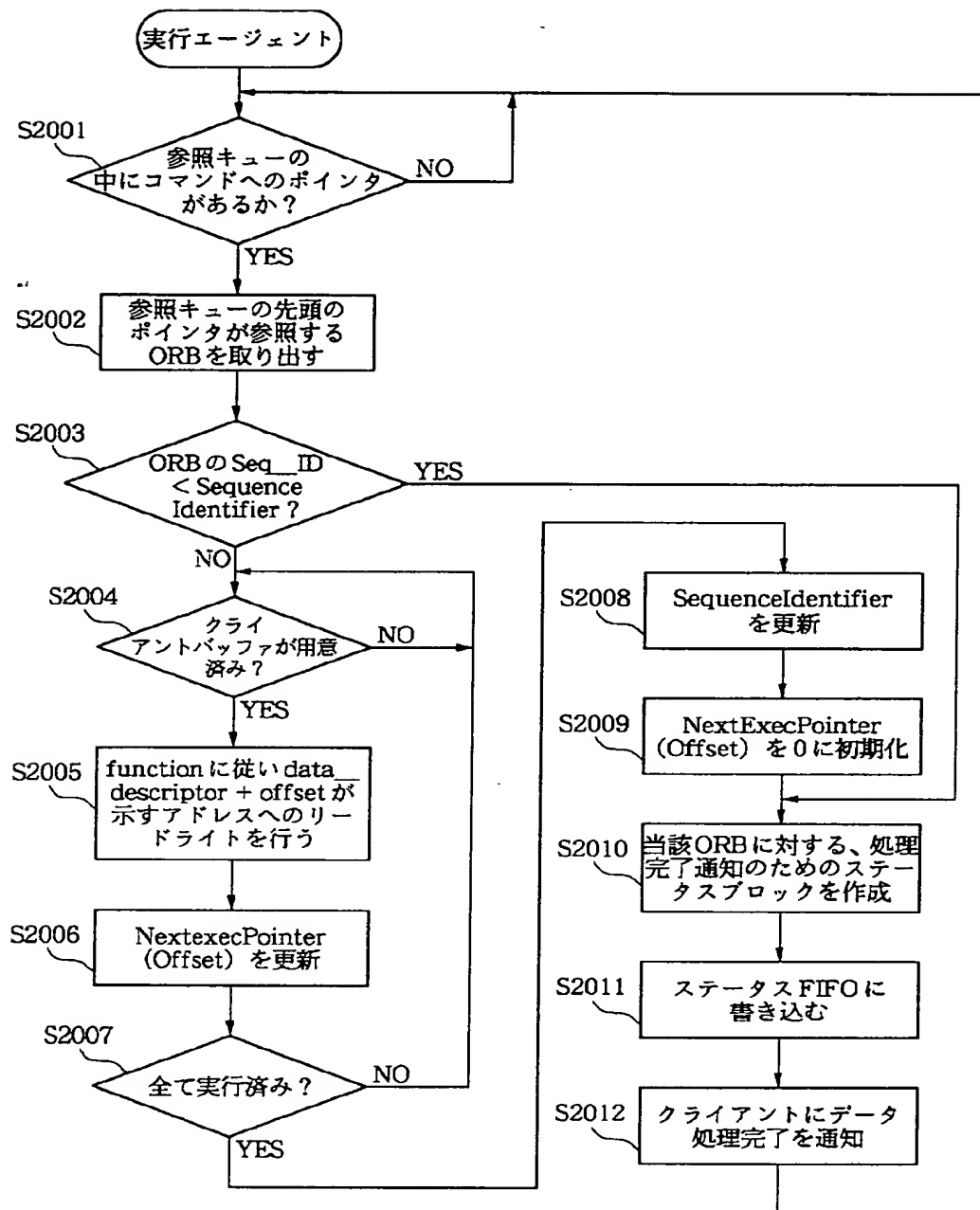


【図 25】

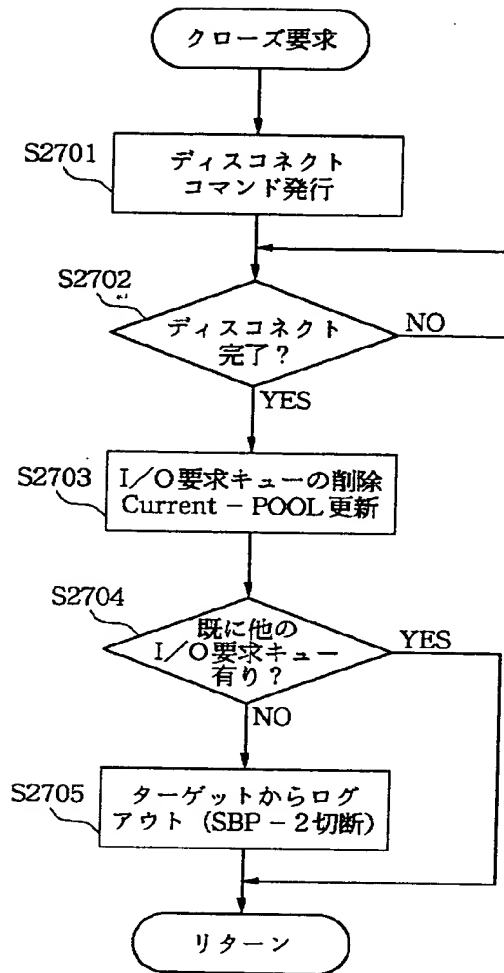


プリフェッチプール容量変更コマンド

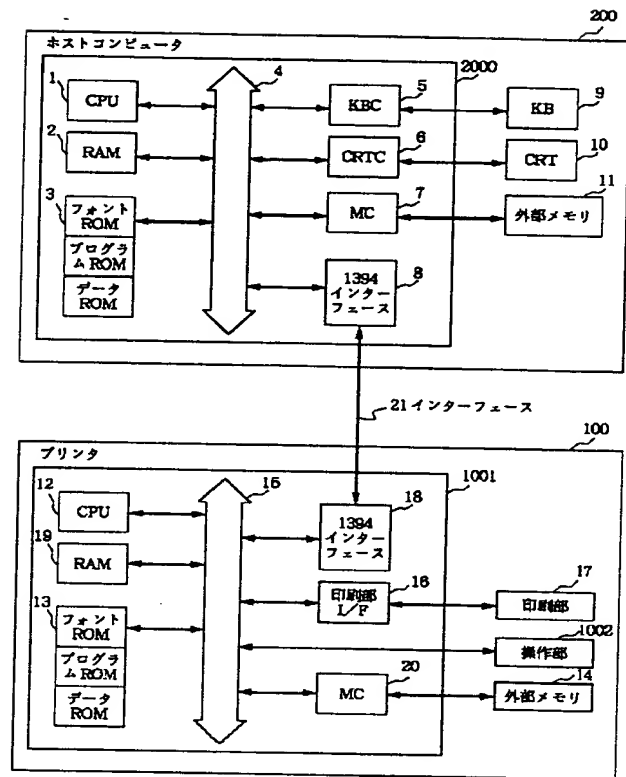
【図 20】



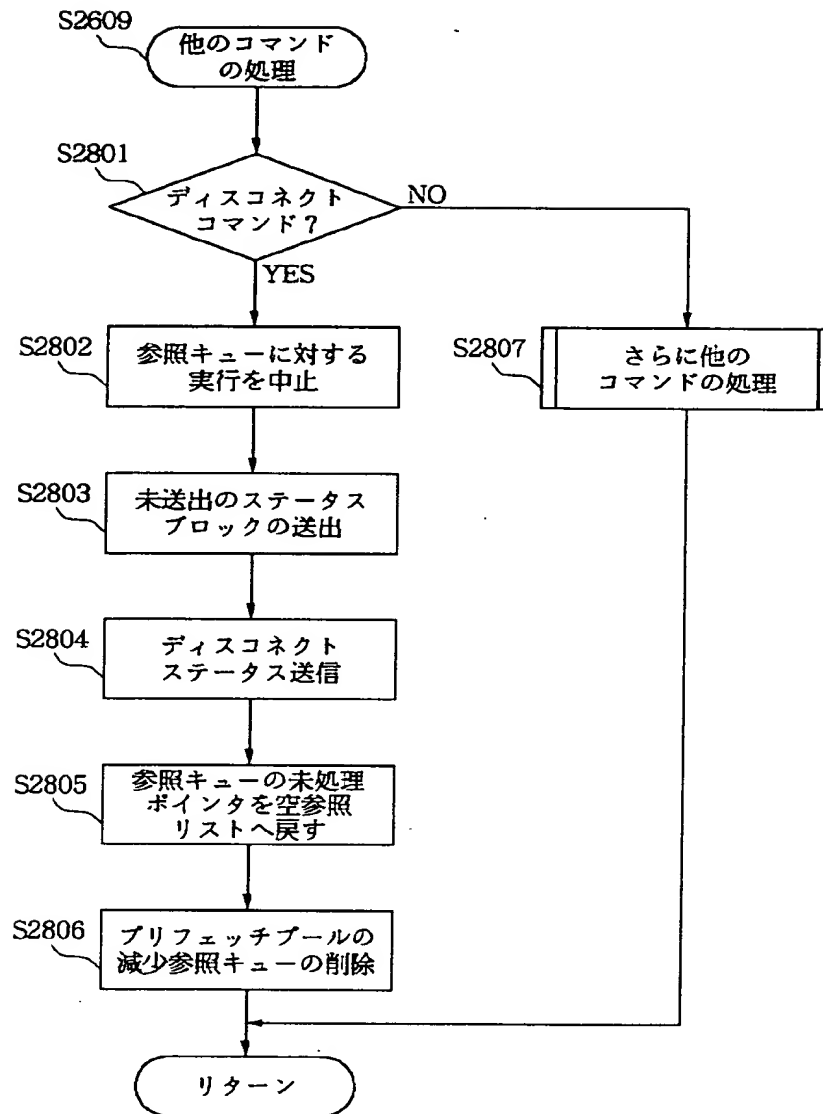
【図 21】



【図 24】



【図 22】



THIS PAGE BLANK (USPTO)